National College of Ireland

Summarizing Newspaper Articles using Optical Character Recognition and Natural Language Processing

MSc Research Project
Data Analytics

Shashank Sanjay Tomar
Student ID: x19213280

School of Computing
National College of Ireland

Supervisor: Dr. Paul Stynes & Dr. Pramod Pathak

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Shashank Sanjay Tomar
Student ID:	x19213280
Programme:	Data Analytics
Year:	2021-2022
Module:	MSc Research Project
Supervisor:	Dr. Paul Stynes & Dr. Pramod Pathak
Submission Due Date:	16/12/2021
Project Title:	Summarizing Newspaper Articles using Optical Character Recognition and Natural Language Processing
Word Count:	7999
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Shashank Sanjay Tomar
Date:	27th January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Summarizing Newspaper Articles using Optical Character Recognition and Natural Language Processing

Shashank Sanjay Tomar
x19213280

Abstract

The present journalism market does not see newspapers as the primary source of information as it once did. In recent years, readers have shifted to more digital and accessible sources, such as social media platforms and messaging applications. Because newspapers are so comprehensive, it is a laborious task to sift through all the information. The key goal of this study is to build an end-to-end solution suite that enables readers to listen to an audio file containing a summary of the articles present in a newspaper, in lieu of reading them. It will attempt to resolve a few long-standing challenges in the field of newspaper digitisation by developing a sophisticated solution capable of handling complex newspaper layouts, lengthy articles, etc. This will benefit the readers by providing a quick and reliable way to consume news by means of audio files. Using an unannotated opensource dataset with scanned pages of a newspaper, a Mask RCNN model was trained to segment the various articles contained within a page. The articles were then taken through another stage of Mask RCNN to identify different text columns in them. After segmenting the column images, Tesseract(OCR) was used to extract the text, which was later put through text cleaning and spell checking using the Microsoft Bing API. To produce a summary of the cleaned text retrieved from the OCR, a second opensource dataset (CNN-DailyMail) was used to train a BERT NLP model. While training on only one fifth of the data used in previous studies, the study produced an effective image segmentation model with a validation MRCNN BBox loss of 0.187 & Mask loss of 0.189 while extracting text from articles with a confidence score of 82.79. Text and audio summaries with a ROUGE-l score of 25.78 and a ROUGE-2 score of 18.21 were also produced.

1 Introduction

Journalism has always provided people with news through multiple sources, including news channels, social media, newspapers, etc. However, one source that has remained essentially unchanged throughout history is newspapers, when it comes to the format as they still are printed and distributed pretty much the same way they have been since their inception. In light of this, one could assume that the entire newspaper industry is quite strong and very unlikely to disappear. Yet, this very industry is declining. In fact it has been reported that daily print newspapers in the USA lost about 20% of their paid subscribers in the period between 2006 and 2011(Pattabhiramaiah et al.; 2018).

This is largely because the new generation consumes news from sources like social media, news channels, etc. Since they are quick and easy to consume, as opposed to reading newspaper articles in their entirety. Citing this decline, researchers have been trying to digitize newspapers to improve access for readers, and thus multiple attempts have been made to digitize newspapers in recent years. As a part of the current state-of-the-art in newspaper digitization, research has been conducted using page scanning, OCR (Optical Character Recognition) and zoning(Klijn; 2008). Efforts have been made, but a comprehensive, end-to-end solution that integrates a large number of sophisticated technologies to digitize newspapers is yet to materialize. Furthermore, the problem of semantically summarizing the articles in the newspaper in order to decrease the time spent reading them remains unresolved.

An imperative motivation for this research was the desire to address the aforementioned gap in previous researches and devise a way for readers to quickly go through the newspaper without spending a great deal of time reading it. Using a novel combination of image segmentation with deep learning, OCR text retrieval, producing an audio summary through NLP, this study will provide a holistic solution for digitizing newspapers. One of the main beneficiaries of this solution will be readers who would like to read the newspaper but lack the time to do so. Instead, they will be able to listen to short summaries of those articles as audio files. Many factors affect the quality of the solution in the research, including, the quality of the scanned pages, the complexity of the page layout, as well as the fluency of the language used in the article, which all eventually influence the quality of an article summary.

The aim of this study is to investigate **"How can Deep Learning, Optical Character Recognition, and Natural Language Processing be integrated and used to summarize newspaper articles?"**. The study aims to address this research question by achieving the following **research objectives**:

- Investigate and implement the current state of the art with regards to digitizing newspapers and producing a summary of news articles.
- Design an end-to-end framework that will analyze a scanned newspaper page and separate the articles in it. After retrieving the text from these article images, generate a text and audio summary for the readers to listen to, allowing them to consume the news more efficiently.
- Implement various components of the framework outlined as:
  - Segment articles from newspaper images using the Mask-RCNN image segmentation model.
  - Using the Tesseract OCR engine, retrieve the article text from segmented image files.
  - By training NLP BERT Text Summarization model, summarize the retrieved article text and produce a summary both in text and audio form.
- Evaluate the results using relevant metrics, such as training loss curves and validation loss curves for image segmentation through Mask-RCN, OCR confidence scores for retrieving text from images and ROUGE Scores for summarizing article text.

A major contribution of this research is addressing the issue of declining newspaper readership caused by readers switching to other news sources due to the lengthy amount of

time required to read a newspaper, by building a framework that uses a novel combination of deep learning (Mask RCNN) trained on a newspaper page dataset[1] from "Times of India" prints from Jan-2018, OCR (Tesseract), and NLP (BERT Model) trained on CNN-DailyMail dataset[2] to produce audio summaries of news articles for swift consumption. Also, the framework's quality will be evaluated using corresponding evaluative metrics, such as loss curves, confidence scores and summary rouge scores.

The rest of the paper focuses on reviewing the related works and literature in Section 2, covering the fields of Image segmentation, Text Recognition and Text Summarization. The research methodology adopted by the study is discussed in Section 3. The design specification of the proposed framework is discussed in Section 4. A description of how the research was implemented follows in Section 5. The evaluation and the experiments performed with their discussion are presented in Section 6 and finally Section 7 concludes the paper by suggesting some possible directions for future research.

2 Related Work

To develop a comprehensive framework for summarizing newspaper articles through newspaper images, we must perform a critical analysis and understand the research that has already been conducted across the various components. This study consists of three main components or subsections, which are **image segmentation** to extract the articles from the newspaper image, **optical character recognition** to retrieve the article's text, and **text summarization** to summarize the retrieved text. Among the abbreviations used in the subsections are Mask RCNN (Mask Region-Based Convolutional Neural Network), OCR (Optical Character Recognition), NLP (Natural Language Processing), ROUGE(Recall-Oriented Understudy for Gisting Evaluation) and BERT(Bidirectional Encoder Representations from Transformers). The purpose of this section is to review prior studies in the aforementioned fields, their findings, and their limitations, and while doing so provide inspiration for this study. This section will have four sub-sections, one for each component, and one for a summary of the learnings from the whole section.

2.1 Image Segmentation of Newspapers

The first step to summarizing news articles and digitizing newspapers is to identify and segment the different articles in the paper. Initially, we examined some of the earliest research into segmenting articles from newspaper images. In a study, Gatos et al. (1999) proposed a rule-based method of identifying the various articles on a page. There are a number of rules to adhere to, including line extraction, image identification, etc., which is quite sophisticated and impressive for the 1990s (according to recall and precision metrics over 96%). It was around the same time when in an another study, Mitchell and Yan (2001) explored a very similar rule-based bottom-up approach to segment articles from newspapers by identifying rectangle blocks called "rects". These methods might have been impressive during those days as newspapers had fairly uniform and strict layouts, but today's newspapers have much more complex layouts and do not only feature rectangular articles like in the past. Unlike today's newspaper layout, where the articles are no longer bounded by just a line, but are also outlined by images, advertisements, etc., such

[1]Times of India Jan-18 Dataset: `https://archive.org/details/TOIDELJAN18`
[2]CNN-DailyMail Dataset: `https://huggingface.co/datasets/cnn_dailymail`

algorithms are very meticulous and do not consider the modern day layout complexity. Our next step was to look at some modern computer vision solutions utilizing deep learning, since rule-based approaches were not sufficiently generic and reliable for our research.

During the quest for a more generalised and complexity-handling approach to segmenting newspaper articles, we came across deep learning-based researches. In one such study, Lee et al. (2020) used 16.3 million historical newspapers between 1789 and 1963 to train a Faster RCNN model to identify the bounding boxes of various components. According to the study, a newspaper page was segmented into seven classes, including a photograph, map, comic, advertisement etc. While achieving a mean average precision of 63.4%. Considering that this research included over 16.3 million newspapers, it made an impressive contribution. The study was limited by the fact that they omitted segmenting the most important part of the article, the textual content, and only looked at images and illustrations, and without the textual content, one can't summarize the article. In addition, since the newspapers used in the study were roughly 2-3 centuries old, they were black and white, had a very simple layout and lacked any complexity. In another study Meier et al. (2017), a Fully Convolutional Neural Network was used in conjunction with some semantic sense in order to segment the components of the newspaper by relying on their relative position within the layout, resulting in a 34.8 fold reduction in runtime (few milliseconds) per page. The idea was inspiring and promising, but the research had several limitations. The dataset contained over 5500 newspapers, but only 426 were labeled correctly, and any image or article not fitting into a rectangle was removed from the training, resulting in a weaker model's ability to identify a wide variety of article shapes in current newspaper layouts. Furthermore, the study simply disapproved the efficacy of OCR for semantic segmentation without even considering how it could be coupled with technologies like NLP to provide some semantic sense to segmentation.

In another study by Almutairi and Almashan (2019), the Faster RCNN model was extended to include a mask identification branch, named the Mask RCNN model. This paper was one of the main motivating papers for our research since it clearly defines the intent, methodology of the experiments and model training in addition to demonstrating impressive results in segmenting articles within a newspaper image. In this study, annotated 750 training pages and 99 validation pages were used. But 50% of them were nothing more than flipped images of the other pages, and so they risk duplication. The Mask RCNN model used in this study was built by Matterport Inc. Despite this study's role in motivating our research, it had gaps and limitations, such as not recognizing non-rectangle articles. The validation loss values (RPN Class Loss: 0.012, RPN Bounding-Box Loss:0.283, Class Loss: 0.202, Bounding-Box Loss:0.12, Mask loss: 0.13) were not distinctly impressive, even with training on a large dataset of 750 annotated images, for around 100 epochs with a high performance V100 GPU, as you will see in subsequent sections, where we achieve somewhat similar values with a much smaller dataset, trained for a smaller number of epochs. The promised results were nevertheless enough to inspire us to study and learn more about Mask RCNN's segmentation capabilities. In their study, He et al. (2017) explain the architecture and performance of Mask RCNN on a variety of tasks such as instance segmentation, human pose detection, etc. In the mentioned approach, a mask is generated while the bounding box of the object is detected simultaneously. Adding a mask prediction branch on top of the Faster RCNN model achieves this without adding a lot of overhead. The study also highlights Mask RCNN's generalization capability, since it outperformed other models such as MNC, FCIS, etc.

on several distinct tasks such as object recognition, instance segmentation, and person identification throughout the COCO dataset. Adaptability to any task or image type just confirms the results of the previous study (Almutairi and Almashan; 2019), making Mask RCNN an excellent choice for the segmentation of article content from newspaper images.

2.2 Text Extracting using Optical Character Recognition (OCR)

Following our research on segmenting article images, the next step was to determine ways to extract all the textual content from those images. Research revealed that there are a number of ways to extract text from an image such as training complex deep learning models or by using a simple OCR engine such as TESSERACT. Showcasing a deep learning solution, Namysl and Konya (2019) present a CNN model that is trained using a multi-document dataset including articles, invoices, and other synthetic documents. Moreover, alpha composting is presented to improve recognition robustness through a novel data augmentation technique. Results were compared to software packages such as ABBYY FineReader & OmniPage Capture (Commercial) and TESSERACT (Open-source). In comparison with others, the proposed deep neural network solution showed very impressive character error rates (0.11-0.16 vs 0.31). While the research was promising, it did pose some drawbacks, such as requiring more time and hardware horsepower to be trained (Nvidia GeForce GTX 745 and Intel Core i7). Additionally, only 16 distinct documents were used to train the model as well as the restriction of training it for each distinct document type separately. As a result, we decided to investigate a simple approach that doesn't rely on model training or fast hardware to extract the text from images.

In a study, Patel et al. (2012) explores and provides a comprehensive look at an open source OCR engine named Tesseract developed by HP, and it discusses the inception and intent behind the tool. Additionally, the study employs the tesseract engine and features an example of how to extract text from a car number plate using the same. Results and features are compared with those of a competing commercial tool called Transym. Transym could manage only 47% of the criteria when it came to colour and grayscale images, while tesseract scored 61% and 70% respectively. In the process, it was substantially faster than transym by over 5 seconds with a very marginal standard deviation of 0.6, indicating the consistency of performance across different samples. In another study (Palekar et al.; 2017), OpenCV and Tesseract were used in conjunction to detect the license plate text in real time. This study emphasizes the importance of image processing prior to processing an image with the tesseract OCR engine. Prior to text extraction by tesseract, several image processing techniques are utilized, such as thresolding, Gaussian blur, erosion, and grayscale scaling. Although both researches only used number plates, they provided a good overview of the engine, even if they lacked diversity in samples. A research by Kaundilya et al. (2019) illustrates the concept of text detection, localisation, segmentation from the background and finally converting it into a binary image by applying text detection together with localisation. To enhance the recognition process, multiple image processing techniques are also employed, such as noise reduction and gray-scale conversion. The aforementioned researches provided us with the goal of selecting tesseract as the text recognition tool, but also propelled us to examine the impact of image quality on recognition.

During their research about tesseract's application in order to extract text from im-

ages, it appeared that many researchers gave considerable consideration to the quality of the images and the image pre-processing since tesseract's results were strongly related to it. Using low resolution images from set top boxes, Brisinello et al. (2017) performed a comparison of tesseract 3.5 and tesseract 4.0 to extract textual information. Using simple preprocessing techniques such as sharpening, binarization, clustering etc., the study reported a bump of 33.3% and 22.6% in performance, respectively. While this improvement was substantial, we believe that key preprocessing techniques such as skew correction could have helped even more. Holley (2009) conducted another comprehensive study aimed at identifying key factors affecting the quality of OCR text output. Factors such as the resolution of the scanned image, the quality of the original source, the bit depth, the skewness of a page and training time (if applicable) are some of the most important determinants of the quality of the resulting text. A positive aspect of the research was that instead of just mentioning the issues, it suggested solutions to improve quality, as well as a method for assessing performance based on the confidence scores.

In order to produce a summary, we intended on combining the retrieved text from images with Natural Language Processing. Therefore, it was essential to learn more about the impact of OCR quality on the downstream NLP tasks. In his study, van Strien et al. (2020) provided the very same information, and the study highlights the challenges resulting from poor OCR results. An analysis of NLP tasks based on OCRed text and NLP tasks based on human-corrected text is presented. In this case, it clearly illustrates how poorly identified texts cause confusion in NLP algorithms and negatively impact summarization. As well as exploring the use of methods such as dictionary lookup to assess OCR quality, the paper suggests that a minimum level of OCR must be at least 80% in order to avoid affecting NLP results.

2.3 Summarizing text using Natural Language Processing (NLP)

Creating a textual summary before creating the audio file was the last but one of the most essential parts of this study. Reviewing the available literature and research in the field of text summarization, we found a study conducted by Allahyari et al. (2017). In this study, the authors address the increase in the volume of textual information in recent years and the need to efficiently summarize it for efficient consumption. A small comparison of abstractive and extractive summarization methods has been presented, showing that extractive summaries are often more accurate than abstractive summaries. It is mainly due to the limitations of the semantic inference capability of any abstractive summary system in existence at present. So, this study emphasizes extractive summarization by using sentence scoring, frequency-driven topics, and sentence selection techniques. Furthermore, the paper also discusses summarization applications using the web, scientific articles, email, graph based and machine learning based summarization. In addition, the study looks at evaluation metrics that can be used to assess the quality of summaries generated using a variety of techniques - human evaluation and automated evaluation methods such as Rouge, a collection of metrics that has been used a lot since the 2000s, such as ROUGE-n, ROUGE-l, etc. Another study by Moratanch and Chitrakala (2017), discusses the utility and effectiveness of extractive text summarization, and discusses two levels of features. There are various word level features such as content, title, cue phrases, upper case, and so on. As well as sentence level features like location, length, and paragraph location. Furthermore, it discusses different supervised (Neural Networks etc.) and unsupervised methods of summarization and explains their benefits and limitations. Similarly

to the previous study (Allahyari et al.; 2017), this research also focused on automated evaluation metrics, such as Rouge, and implied their importance and effectiveness.

After reviewing the above studies, we decided to explore the use of deep learning methods to create a summary. An application of pretrained encoders in text summarization was demonstrated in a study by Liu and Lapata (2019). One of the most promising models across NLP tasks has been Bidirectional Encoder Representations from Transformers (BERT). The study trains a BERT model using three distinct datasets(CNN-DailyMail, NYT, XSum) and suggests a more generalised model that can accommodate both extractive and abstractive summary models. Based on the comparison of the generated summaries, BERT was found to outperform all other models across all datasets. The comparison was made with Rouge metrics as in previous studies. In another study, Miller (2019) emphasizes that BERT not only takes into account the syntactical aspects of the text, but also heavily relies on the semantic context of the text in order to produce a summary. An emphasis of the study is on developing a lecture summarizing service, which uses the BERT model to create textual embeddings. Additionally, k-means clustering was used to narrow down to the centroid aligned sentences that ultimately made the summary. Moreover, the study revealed that traditional extractive summarization methods ignored context and produced a very poor result that required a great deal of manual intervention. It took an example of such dated methodology such as TextRank to demonstrate how much better a summary generated by the BERT model is compared to one generated by TextRank. While describing their solution, they also refer to its limitations, such as the inability to handle a certain set of words, such as "this", "also", etc. as the model kept on searching for a further context, and suggests libraries like NLTK to remove such words. Moreover, their text was predominantly conversational (lecture-based), which made context identification even more important. According to a study by Tenney et al. (2019), pre-trained encoder models have shown great promise across different NLP tasks in recent times, supporting the findings of the aforementioned literatures. In a way, they illustrate how BERT is akin to the conventional NLP pipeline, from POS tagging to all the way to coreference, and its ability to alter the order of stages in this pipeline. There is a strong inclination towards advocating BERT as a sophisticated approach to text summarization across all of the aforementioned studies.

2.4 Summary of Related Work

Based on our review, we conclude that there is a common inclination towards certain methodologies or approaches in the fields of image segmentation, text extraction, and text summarization. When identifying and segmenting newspaper articles, studies have demonstrated better results with deeper learning solutions compared to rule-based ones. As a result, we learned that Mask RCNN can be used for semantic article segmentation by identifying masks and boundary boxes. On the other hand, Tesseract has proved to be a better, faster and more efficient choice for recognizing text, especially considering that it is an open source tool. Quality and pre-processing of images were also identified as important factors for improved results. When summarizing the retrieved text, deep learning solutions such as a pre-trained encoder model such as BERT were efficient due to their ability to take into account the underlying semantic context. Additionally, ROUGE was frequently mentioned as a tried-and-tested metric for evaluating the summary.

The research reviewed here, however, had some or the other drawbacks, such as not being able to identify and segment non-rectangle articles and requiring a large dataset

of newspaper images in order to train the model, lack of consideration of semantics (no grammar/spelling check) within the generated output from OCR, etc. This study aims to address these gaps and issues and many more by presenting a novel solution and framework for efficiently summarizing articles from newspaper images into text and audio files.

3 Methodology

This section provides an overview of the steps involved in conducting the research and the methodology adopted. When designing the methodology for this research, several aspects of the CRISP-DM methodology were used as inspiration as it is one of the most prominently followed methodologies. The research primarily follows 5 distinct steps such as data collection, pre-procesing, transformation, modelling and evaluation as can be inferred from the figure 1.

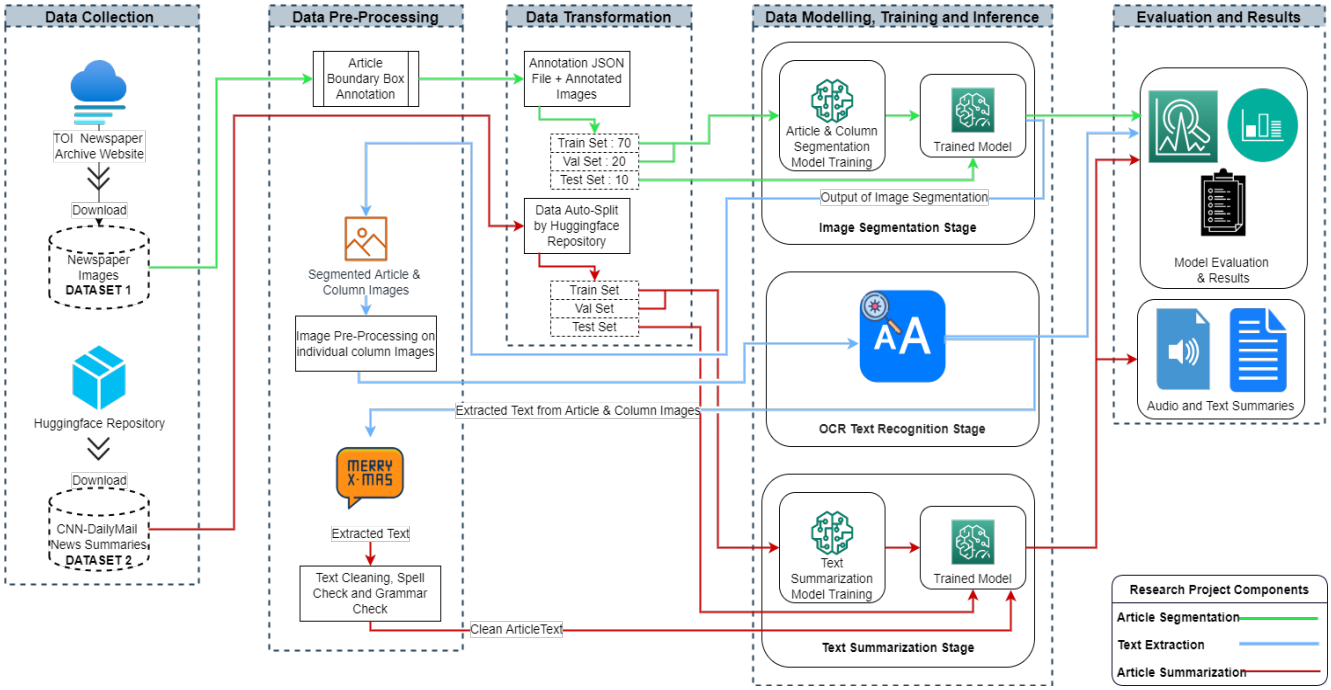

Figure 1: Research Methodology

The first step in the data collection process was to identify the data sources of newspaper image dataset and news article-summary pair dataset. These scanned newspaper images were downloaded from the online archive data repository[3] for "Times of India-Jan 18". The archive consists of over 1300 scanned images containing articles and images. However, many of them had only pictures and promotional prints, and did not have any articles, so they were excluded from the final dataset. Using huggingface's Python package, the second dataset[4] CNN-DailyMail, containing over 311971 pairs of article-summary pairs was also downloaded.

The next step was data pre-processing, and as can be seen in the figure 1, since the research dealt with multiple stages of image segmentation, text recognition, and text

[3]Times of India Jan-18 Dataset: `https://archive.org/details/TOIDELJAN18`
[4]CNN-DailyMail Dataset: `https://huggingface.co/datasets/cnn_dailymail`

summarization, the output of these stages was considered as an input for the next one, therefore pre-processing was required at both ends. First, we manually annotated(2 classes: Rectangle and Non-Rectangle Articles) over 166 newspaper images from the original dataset in order to leverage as few images as possible to train the model and still obtain decent results when identifying article segments. A second round of pre-processing techniques, such as gray scaling and thresholding, was applied once the segmented article images were obtained. As the final stage of pre-processing, the extracted text from those images was cleaned up, spell checked, and grammar checked.

During the third step of data transformation, annotations of newspaper images were converted to a JSON file to match the requirement of the mask and boundary detection models. An annotation tool called VGG image annotator[5] helped us achieve this goal by providing us with regions, classes, and x,y coordinates of the annotations. The image dataset, along with the JSON annotations, was then split into 70:20:10 ratios as train, validation, and test sets. As for the second dataset, which contained article-summary pairs, no transformations were required and was already provided with the train, validation, and test split.

In the fourth step of the methodology, various models were trained and inferences were made. A sequence of three main stages was followed. The first stage involved training a deep learning neural network using transfer learning, based on Mask-RCNN to identify masks and bounding boxes of articles from newspaper images. In the second stage, the text was extracted from the identified article images through the use of optical character recognition by simply using Tesseract OCR's base pre-trained engine. As the final step, train a NLP model based on BERT to semantically summarize the extracted textual information from the articles. Detailed descriptions of all the frameworks and models outlined above will be discussed in detail in the next section focusing on design specification.

In the fifth and final evaluation step, the performances of all models were evaluated using several metrics such as the training and validation loss curves for image segmentation, the OCR confidence scores for text extraction, and ROUGE metrics such as Rouge-n, Rouge-l, etc. for summary evaluation. A text and audio file was created from the final summary of the articles for easy consumption.

4 Design Specification

This section provides insight into the architecture and frameworks used to conduct the research and sheds light on the different components. On the basis of the figure 1, the modelling stage of the methodology illustrates three major components of the study, namely, image segmentation, text extraction, and text summarization. The figure 2 provides a comprehensive view of the internal architecture of these components.

4.1 Image segmentation using MaskRCNN

In the figure 2, the first component identifies and segments different articles in the newspaper image. We benefited from the power of transfer learning and used a Mask RCNN model that was built using ResNet101 and FPN and pre-trained on the COCO dataset as our basis. Then we train a custom Mask RCNN model using the manually annotated

[5]VGG Image Annotator: `https://www.robots.ox.ac.uk/~vgg/software/via/`

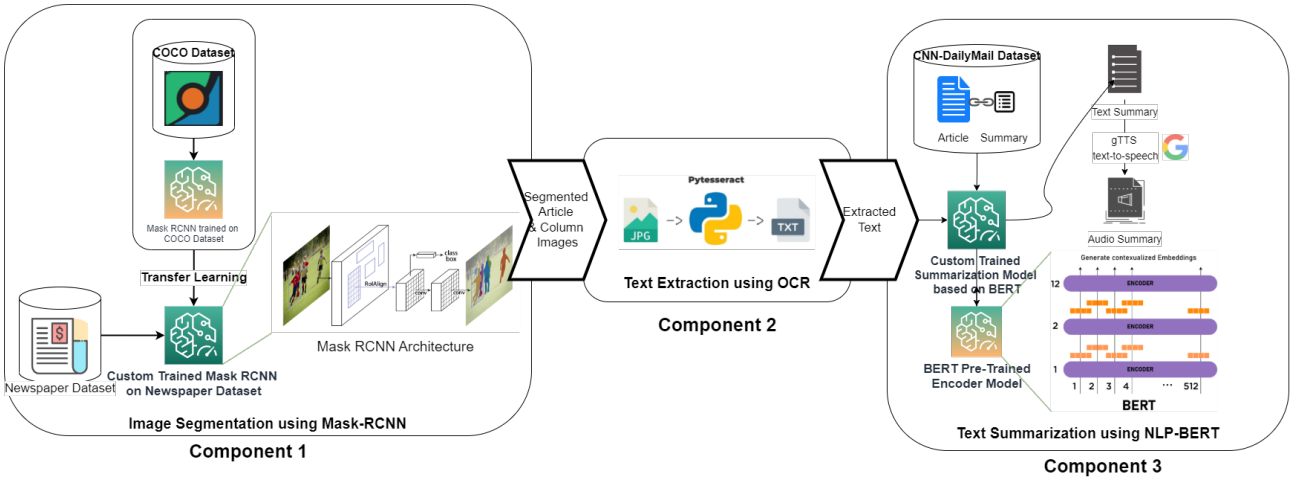

Figure 2: Framework Architecture of Research

images from newspaper page dataset. It also provides a glimpse into the inner architecture of the Mask RCNN model, which is basically a Faster CNN with an extra branch on top to identify the mask of the object being detected along with the boundary boxes and class as well. This ability of identifying the interested region in the form of a mask on the object, helped us identify non-rectangle articles as well. It is important to note that in an attempt to improve downstream OCR, the Mask RCNN model was trained twice, first on the entire newspaper to identify article segments and then on these segmented article pictures to identify different columns within each article. Once the mask and bounding box were identified, the newspaper image was cropped into multiple articles and columns within those articles according to x, y, width and height specifications.

4.2 Text Extraction using Tesseract OCR

The second component involved extracting text from the cropped images of columns within an article. Articles were cropped into columns in the previous component because Tesseract OCR reads lines by lines, and horizontal adjacent text lines would be confusing to produce semantically correct output. In order to extract text from all these cropped column images, we used the PyTesseract package, which operates by reading images and extracting text from them using a pre-trained Tesseract engine. Basically, the package wraps the Tesseract engine initially developed by HP and now owned by Google, which is used by them to identify spam emails etc. In its base form, it can identify over 100 languages with its text recognition capabilities and can be trained to identify more. We did not need to train it any further since our newspaper was in the "English" language. We concatenated the extracted text from the columns to create a single text document containing all of the article's content.

4.3 Text Summarization using BERT-NLP

In the third component of the trilogy, a summary was to be produced from the OCRed text. Based on the article-summary pairs in the CNN-DailyMail dataset, a custom BERT deep learning model was trained. Bidirectional Encoder Representations from Transformers (BERT) as the name implies is a pre-trained bidirectional encoder transformer that was outlined for the first time in a study conducted by Devlin et al. (2018). The

model was initially trained with the goal of improving contextual understanding of models when performing NLP tasks. BERT's base version explores the notion of next sentence prediction and efficiently identifies tokens that are masked. Based on its capabilities, we chose to train our custom model based on the BERT-base model which has 12 transformer layers and a maximum of 512 tokens per sentence, as can be seen in a glimpse of the BERT architecture in the figure 2. After an extracted text was generated from the previous component and subsequently cleaned, this custom BERT-NLP Summary model generated a summary, which was eventually converted into an audio file.

5 Implementation

This figure 3 shows an overview of the lifecycle of implementing this research. We used Python 3.7 as well as Jupyter Notebooks and Google Colab Pro as key IDEs for implementing the research. While running Jupyter Notebook, a Ryzen 4900h CPU and a RTX 2060 MaxQ GPU were used, while Google Colab Pro used a P100 GPU.

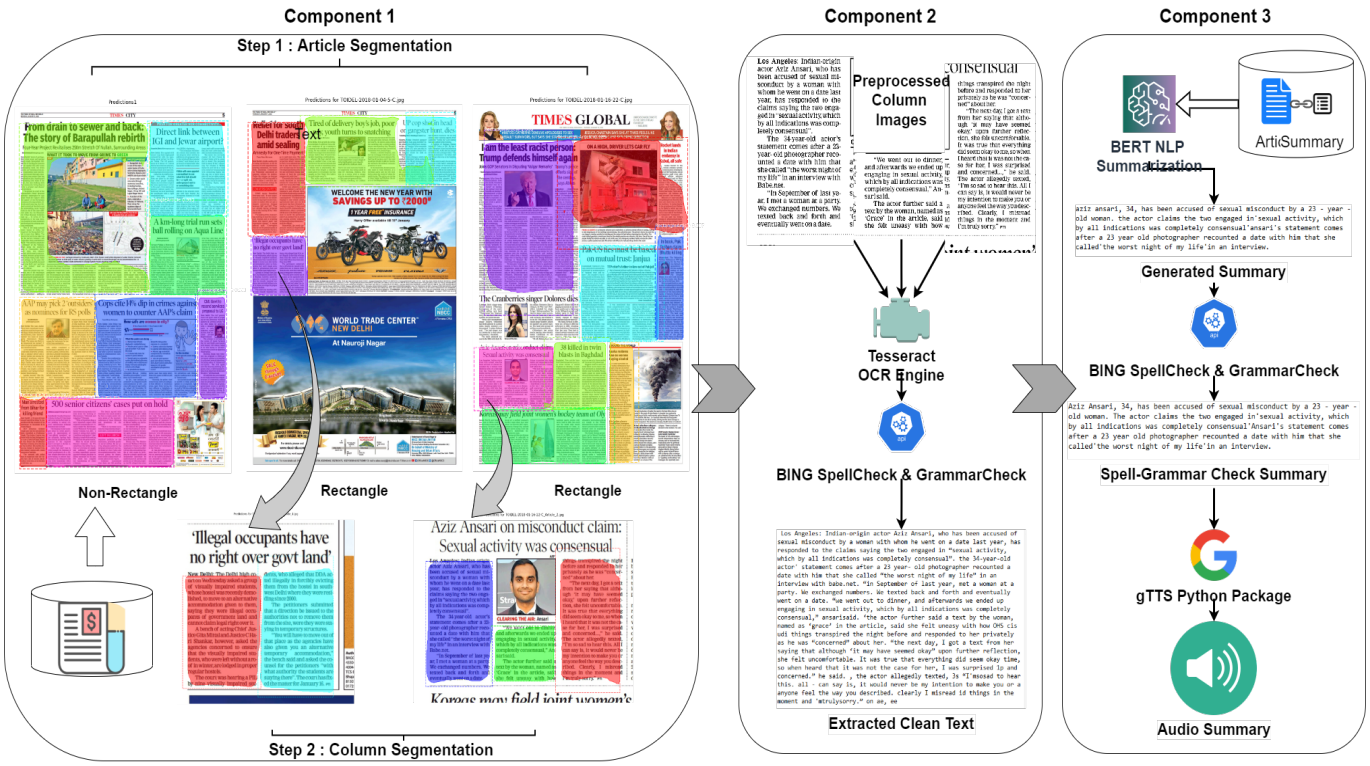

Figure 3: Research Implementation Lifecycle

An initial step of the first component was to manually label the Time of India newspaper images into two categories (Rectangle and Non-Rectangle Articles) using VGG image annotator by marking the bounding boxes of articles. The result of this was the creation of a JSON file for training the image segmenting Mask RCNN model using transfer learning based on weights of a Mask RCNN model trained on COCO Dateset. Once the model had been trained and the boundary boxes identified, OpenCV was used to crop images of the article columns.

With regard to the second component, the cropped column images were subjected to image processing, such as grayscaling and thresolding using the OpenCV library. Using

the PyTesseract Library, we subsequently extracted text from these images and consolidated it to form the textual content of one article. Using the NeatText library and some in-house developed code, the extracted text was cleaned by removing special characters, usernames, etc. We then used the Microsoft Bing API to do a spelling and grammar check to ensure that the text had some semantic meaning for the downstream NLP tasks. Both the extracted text and the cleaned version were saved separately.

In order to summarize these clean texts, the final third component was built. The dataset was downloaded directly from the huggingface "datasets" library since the repository is accessible without any extra downloads through the library. Due to the large number (311971) of article-summary pairs in this CNN-DailyMail dataset, we needed a better GPU to train the model, so Google Colab Pro was employed. Dataset were downloaded into the notebook's cache on colab. Using "transformers", "torch" libraries we trained the custom BERT NLP model based on the "bert-base-uncased model" with the acquired dataset. The performance of this model was then evaluate using Rouge Library to identify the Rouge metrics. Following the generation of the summary from the trained model, it was rechecked for spelling and grammar. Eventually, to generate the audio file of the summary, a Google text to speech interface was utilized by using the gTTS library.

The Table 1 shows a clear distribution of tools, languages, libraries, models etc. used.

Table 1: Tools, Languages, Models and Libraries used in research

Description	**Name**
Language	Python 3.7
IDEs	Jupyter Notebook, Google Colab Pro
Component 1	
Image Annotations	VGG Image Annotator
Base Weight for MaskRCNN Training	MaskRCNN trained on COCO
Displaying Mask and BBox	Matplotlib
Train-Loss Validation Curves Evaluation	TensorBoard
Component 2	
Image Processing	OpenCV
Text Extraction	PyTesseract
Text Cleaning	NeatText
Spelling-Grammar Check	Microsoft BING API
Component 3	
Dataset Access	"Datasets" by huggingface
Base model for BERT-NLP	bert-base-uncased
Text to Speech	gTTS (Google Interface)
Summary Evaluation	Rouge Package
Miscellaneous Libraries	
	NumPy, Shutil, JSON, Requests
	Math, Statistics, Regex etc.

6 Evaluation

Following the implementation of the research, all experiments conducted in the study were evaluated. Three major experiments were conducted for each component of the

study, and their performance was evaluated using a set of corresponding metrics. In the first experiment (Subsection 6.1), newspaper images are segmented into different article columns using MaskRCNN. The second experiment (Subsection 6.2) extracts text content from these images using Tesseract. The final experiment (Subsection 6.3) generates an audio summary from the extracted text, using BERT-NLP. In the Subsection 6.4, we discuss the learnings and inferences derived from the following experiments.

6.1 Experiment 1 : Article and Column Segmentation using MaskRCNN

In this experiment, the aim was to recognize and segment different articles, and then to segment them again into different columns. In order to do so, 166 newspaper images were annotated, and the reason for such a small dataset was that the study wanted to build a lightweight yet efficient segmentation model that is easy to train and yet is an effective tool. This efficiency will be evident in the results discussed ahead. For our base weights, we used the same MaskRCNN model (COCO) used by Almutairi and Almashan (2019), which was built by Matterport Inc. According to subsection 4.1 above, we had to train two MaskRCNN models to segment articles and columns in a two-step process. We found that the best results were achieved when the batch size was set to 2, the number of steps to be 131 for the first article segmentation model and 52 for the second column segmentation model. In both models, 20 epochs were trained, and a detection confidence level of 85% was set. However, while cropping the image later on, using the inference of these models, the detection confidence level was increased to 95% to produce more accurate segmentations. Figures 4 and 5 show the loss curves for the First Model (Article Segmentation) and the Second Model (Column Segmentation).

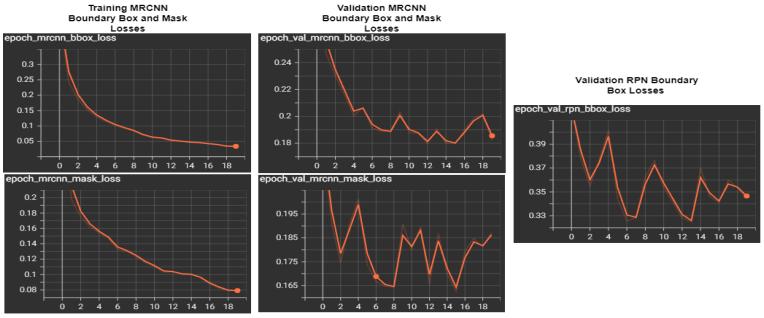

Figure 4: First MaskRCNN Model(Article Segmentation) Loss Curves

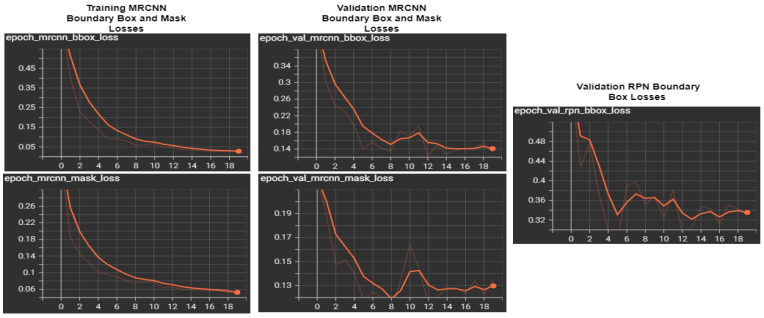

Figure 5: Second MaskRCNN Model(Column Segmentation) Loss Curves

According to Figure 4, the first model does a good job of identifying masks as well as boundary boxes of the articles during both training and validation. As the number of epochs increases, the Bounding Box and Mask loss in training decreases, but this is not the case in validation. In comparison, Validation MaskRCNN Bounding Box Loss decreases more linearly than Validation MaskRCNN mask loss. In addition, we analyzed the validation RPN loss, which showed just the Boundary Box identification loss. The fact that boundary box identification was crucial to cropping the image into various articles led us to focus on epochs 12 and 16, but a lower loss of validation mask detection at epoch 12 encouraged us to use it as our final model. The validation loss values of the model at 12 epochs were MRCNN BBox Loss 0.1872, MRCNN Mask Loss 0.1897, and RPN BBox Loss 0.3, which will be compared with the state of the art solution by Almutairi and Almashan (2019) in the discussion section 6.4. Based on figure 5, the second model also does a good job of identifying and segmenting columns from the article images. Identifying boundary boxes and masks, both, here results in a linear decrease in the validation loss. Despite a slight increase in mask identification loss values after epoch 8, the loss values for Boundary box identifications keep decreasing all the way till epoch 20, which is also confirmed by looking at the validation RPN Boundary box loss curve. We therefore decide to use the model at epoch 20 as our final model. MRCNN BBox Loss 0.13, MRCNN Mask Loss 0.13, and RPN BBox Loss 0.33 were the validation loss values at 20 epochs.

The losses of both the models demonstrate the effectiveness of MaskRCNN in segmenting articles and columns. In particular, the fact that the loss values were almost the same for mask and boundary box identification shows consistency as well as a smaller positional gap between the mask and corresponding boundary box.

6.2 Experiment 2 : Text Extraction using Tesseract

In the second experiment, the goal was to extract the text from the segmented article and column images. To do so, we used the pre-trained TESSERACT engine as the language we needed to identify was "English" and no explicit training was required. However, we did perform some image processing and compared the performance of Tesseract between processed and unprocessed images. The figure 6 below illustrates an example of image processing using OpenCV. The extracted text was also checked for spelling and grammar mistakes using a Microsoft BING API. We evaluated the performance of the text recognition based on OCR confidence score and the number of changes suggested by the API.

Figure 6: Example of Image Pre-Processing

Detailed information about the confidence scores of Tesseract's performance and the number of changes suggested by the SpellCheck API can be found in the Table 2.

Table 2: Tesseract Performance Metrics

Image Type	Confidence Score	SpellCheck Changes(No.)
Un-Processed	81.4251	4902
Processed	82.7950	4802

According to the Table 2, the confidence score for a processed image is only slightly higher. Also, the number of suggested changes from the Spell check API are slightly better in the processed version. As demonstrated, quality improvement and pre-processing can positively affect Tesseract's performance and improve text recognition. There was just a slim improvement in this case because the quality of the newspaper images (over 2 Mb per image) was pretty good, right from the beginning, that is why the preprocessing didn't improve the recognition of text by a large margin, but it was positive nevertheless.

6.3 Experiment 3 : Text Summarization by BERT-NLP

With this experiment, the aim was to create a textual summary and even an audio summary based on the text retrieved in the previous experiment. As a base model for training our custom BERT-NLP Summarization model, we used the Bert-base-uncased transformer model designed by Devlin et al. (2018). At the original source, CNN-Dailymail's dataset had already been cleaned and split. For our custom BERT-NLP summarization model, the parameters that produced the best results were a batch size of 16, a maximum encoder length of 512, and a maximum decoder length of 128. As part of the evaluation strategy based on "Steps", a validation check was conducted every 8000 to continuously improve the model. During training, the model ran over 53835 steps due to the batch size of 16. Figure 7 shows a detailed description of the validation steps and the associated metrics.

Step	Training Loss	Validation Loss	Rouge2 Precision	Rouge2 Recall	Rouge2 Fmeasure	Runtime	Samples Per Second
8000	2.959800	2.891456	0.098700	0.145800	0.113700	525.446900	2.545000
16000	2.611700	2.634151	0.104400	0.155300	0.120700	534.901200	2.500000
24000	2.359700	2.511087	0.110000	0.163100	0.127200	544.302700	2.456000
32000	2.293400	2.437275	0.107100	0.159900	0.124100	563.116300	2.374000
40000	2.105600	2.394423	0.110400	0.161700	0.127100	545.211800	2.452000
48000	2.069000	2.363725	0.106700	0.156600	0.122800	557.875000	2.397000

```
TrainOutput(global_step=53835, training_loss=2.5525387110390834, metrics={'train_runtime': 87112.8614
```

Figure 7: BERT-NLP Model Training

In order to evaluate the performance of the summaries, we used ROUGE Metrics as suggested by multiple studies by Moratanch and Chitrakala (2017) and Allahyari et al. (2017). We ran the model on the test split of the CNN-Dailymail dataset and obtained the various ROUGE Metrics as shown in Figure 8. To measure the effectiveness of a model, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) compares the reference summary to the summary generated by the model.

Rouge-1 Scores measure the overlap of unigrams between the summaries, which is expressed as the ratio of single words overlapping. In a similar way, Rouge-2 signifies

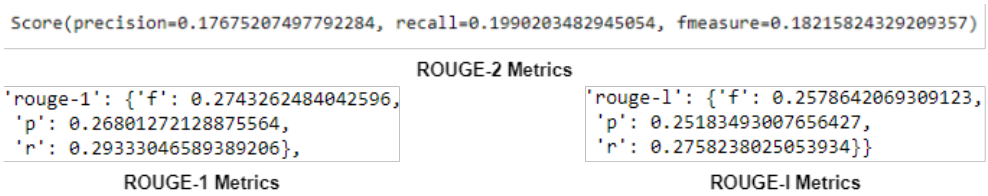

Figure 8: ROUGE-2, ROUGE-1 and ROUGE-l Metrics

the overlap of bigrams, or the ratio of two consecutive words overlapped between two summaries. Despite the fact that the CNN-Dailymail Dataset reference summaries were produced by professional journalists, our BERT-NLP model produces decent precision, recall, and f-measure for ROUGE-2 (17.6,19.9,18.2) and ROUGE-1 (26.8,29.3,27.4). A third metric, ROUGE-l (25.1,27.5,25.7), was also calculated to illustrate the efficient performance of this summary generation model since it does not rely on n-grams in consecutive order.

6.4 Discussion

Several inferences and learnings have been drawn from evaluating all the experiments and models developed during this research. As far as we have been able to determine while conducting the first experiment, this is the first study to apply MaskRCNN to both identify and segment articles in a newspaper image and then segment those article images into columns. In addition to improving downstream OCR, this was mainly done to improve the semantic sense of the extracted text, since Tesseract reads line by line. Past attempts have been made to segment articles, such as a study by Almutairi and Almashan (2019). However, the major drawbacks have been the inability to identify non-rectangle articles and the lack of a two-stage MaskRCNN segmentation that can identify text from individual columns. Both of these concerns were addressed in this study, which demonstrated its ability to identify rectangle and non-rectangle articles along with column identification. In comparison, a study by Almutairi and Almashan (2019) did show lower Validation losses, but the model was trained on five times the data (750 vs 166(our data)), so it would have taken a significant amount of time and hardware. A Table 3 comparing the losses can be found below. As you can see, the loss values displayed aren't drastically different even when there is a five fold difference in the amount of data. This illustrates that one doesn't necessarily need a large number of annotated images to produce an efficient image segmentation model.

Table 3: MaskRCNN Validation Loss Comparison with Previous Studies

Loss	**Our Model**	**Almutairi and Almashan's Model**
RPN BBox Loss	0.3	0.28
Mask RCNN BBox Loss	0.18	0.12
Mask RCNN Mask Loss	0.18	0.13

As part of the text extraction phase, a novel technique for spelling and grammar checking was introduced using the BING SpellCheck API. This was primarily done to maintain the semantic meaning of the article, which could then be used to generate a logical summary by the downstream BERT-NLP Summary generator model. The

evaluation results also confirm the findings of previous research by Brisinello et al. (2017) on image pre-processing, such as that emphasized the importance of image quality in text recognition. Furthermore, in addition to determining the quality of OCR by a confidence score, this study used Spell check to identify how many changes were suggested, since a large number of suggestions would indicate poor OCR quality. As suggested by the van Strien et al. (2020), we achieve a confidence score of 82.79 (more than 80) to limit the impact that poor OCRed text can have on downstream NLP tasks.

The study confirms and acknowledges BERT's capabilities in NLP tasks and its ability to consider the context and meaning within a text while processing it, as suggested by Miller (2019), Devlin et al. (2018) and Liu and Lapata (2019) within past researches. Additionally, it added a spellcheck to the summary generated by the BERT-NLP in order to reduce spelling and grammar errors. In addition to this, the study produces a ROUGE-L score of 25.78, which is very impressive when you consider that as the text length increases, the ROUGE scores tend to decline since it is essentially a measure of the degree of overlap between reference text (usually corrected by humans) and model generated text.

However, there were also some limitations in this study that can be considered as a potential area for improvement. Such as occasionally, the MaskRCNN model would identify an article in two distinct bounding boxes, which overlapped when the inference detection confidence level was lower than 95%. In order to address this drawback, the quality and quantity of annotations could have been improved if there were no time limitations. As a result of using "justification" alignment and having variable space lengths between words in newspaper, poor OCR text recognition also presented a limitation when words were too closely printed on the newspaper and the spaces between them couldn't be identified. Another drawback was the lack of an alternative to ROUGE evaluation to evaluate the summaries. ROUGE heavily relies on only syntactical matching and would only properly assess a text's similarity if the exact words are same without considering its semantic meaning. Reviewing the summaries with human involvement could serve as an additional verification step in addressing this issue.

7 Conclusion and Future Work

The question addressed in this research was to determine how deep learning, optical character recognition, and natural language processing can be used to create an audio and text summary of newspaper articles. Additionally, the prime research objective was to implement a framework that uses MaskRCNN for segmenting articles from a newspaper, Tesseract OCR engine to extract text from article images and then BERT-NLP Text Summarization model to produce a text and audio summary. Ultimately, evaluate the framework components by using validation loss values, OCR confidence score and ROUGE score. As demonstrated by the results and evaluations, the study largely achieved its objective. Based on the results, MaskRCNN bounding box loss was 0.187, text recognition confidence score was 82.79, and summary generation ROUGE-l score was 25.78. One of the key findings was the creation of a framework that demonstrated the potential of using these technologies to generate news article summaries. Another key finding of the study was the creation of a method for digitizing newspapers that was efficient and extremely reliable, such that it could even segment columns within an article. However, this study had some limitations as well, among them were overlapping

of article bounding box detection (for inference confidence detection levels less than 95%) and a lack of semantic criteria for gauging summary performance etc.

Considering the research was focused on fields with a high level of activity, there is a lot of potential to expand the research in the future. There is a scope of improving the annotation quality and quantity for the segmentation model along with adding a multilingual dataset of newspaper images instead of only focusing on the "English" language. It is also possible to improve the pre-processing of newspaper images so the framework identifies the spaces between words better. In order to generalize the framework, images from more newspaper publishers can be used since each has a different structure and level of complexity. On the commercial front, this framework may be used as a backbone for a mobile application which allows the user to snap a picture of any newspaper and listen to a summary of the articles included in it.

References

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B. and Kochut, K. (2017). Text summarization techniques: a brief survey, *arXiv preprint arXiv:1707.02268* .

Almutairi, A. and Almashan, M. (2019). Instance segmentation of newspaper elements using mask r-cnn, *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, IEEE, pp. 1371–1375.

Brisinello, M., Grbić, R., Pul, M. and Anđelić, T. (2017). Improving optical character recognition performance for low quality images, *2017 International Symposium ELMAR*, IEEE, pp. 167–171.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* .

Gatos, B., Mantzaris, S., Chandrinos, K., Tsigris, A. and Perantonis, S. J. (1999). Integrated algorithms for newspaper page decomposition and article tracking, *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, IEEE, pp. 559–562.

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017). Mask r-cnn, *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Holley, R. (2009). How good can it get? analysing and improving ocr accuracy in large scale historic newspaper digitisation programs, *D-Lib Magazine* **15**(3/4).

Kaundilya, C., Chawla, D. and Chopra, Y. (2019). Automated text extraction from images using ocr system, *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, pp. 145–150.

Klijn, E. (2008). The current state-of-art in newspaper digitization, *D-Lib Magazine* **14**(2).

Lee, B. C. G., Mears, J., Jakeway, E., Ferriter, M., Adams, C., Yarasavage, N., Thomas, D., Zwaard, K. and Weld, D. S. (2020). The newspaper navigator dataset: Extracting and analyzing visual content from 16 million historic newspaper pages in chronicling america, *arXiv preprint arXiv:2005.01583* .

Liu, Y. and Lapata, M. (2019). Text summarization with pretrained encoders, *arXiv preprint arXiv:1908.08345* .

Meier, B., Stadelmann, T., Stampfli, J., Arnold, M. and Cieliebak, M. (2017). Fully convolutional neural networks for newspaper article segmentation, *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1, IEEE, pp. 414–419.

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures, *arXiv preprint arXiv:1906.04165* .

Mitchell, P. E. and Yan, H. (2001). Newspaper document analysis featuring connected line segmentation, *Proceedings of sixth international conference on document analysis and recognition*, IEEE, pp. 1181–1185.

Moratanch, N. and Chitrakala, S. (2017). A survey on extractive text summarization, *2017 international conference on computer, communication and signal processing (ICCCSP)*, IEEE, pp. 1–6.

Namysl, M. and Konya, I. (2019). Efficient, lexicon-free ocr using deep learning, *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, pp. 295–301.

Palekar, R. R., Parab, S. U., Parikh, D. P. and Kamble, V. N. (2017). Real time license plate detection using opencv and tesseract, *2017 international conference on communication and signal processing (ICCSP)*, IEEE, pp. 2111–2115.

Patel, C., Patel, A. and Patel, D. (2012). Optical character recognition by open source ocr tool tesseract: A case study, *International Journal of Computer Applications* **55**(10): 50–56.

Pattabhiramaiah, A., Sriram, S. and Sridhar, S. (2018). Rising prices under declining preferences: The case of the us print newspaper industry, *Marketing Science* **37**(1): 97–122.

Tenney, I., Das, D. and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline, *arXiv preprint arXiv:1905.05950* .

van Strien, D., Beelen, K., Ardanuy, M. C., Hosseini, K., McGillivray, B. and Colavizza, G. (2020). Assessing the impact of ocr quality on downstream nlp tasks.