DIGITIZATION OF ANSWER SCRIPTS USING SEMANTIC SEGMENTATION

A PROJECT REPORT

Submitted By

CHARULATHA S	**195002024**
DUVICKSHA U	**195002035**
POOJA LAXMI S	**195002081**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

Department of Information Technology

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)
Kalavakkam - 603110

May 2023

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report titled **“DIGITIZATION OF ANSWER SCRIPTS USING SEMANTIC SEGMENTATION”** is the *bonafide* work of “**CHARULATHA S (195002024)**, **DUVICKSHA U(195002035)**, and **POOJA LAXMI S (195002081)**” who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. C. Aravindan	**Dr. C. Aravindan**
Professor & Head of the Department	**Professor & Head of the Department**
Department of Information Technology,	Department of Information Technology,
SSN College of Engineering,	SSN College of Engineering,
Kalavakkam - 603 110	Kalavakkam - 603 110

Submitted for the examination held on. . . . .

Internal Examiner	**External Examiner**

ACKNOWLEDGEMENT

I thank GOD, the almighty, for giving me strength and knowledge to do this project.

I would like to thank and have a deep sense of gratitude to my guide **Dr. ARAVINDAN CHANDRABOSE**, Professor and head of the Department of Information Technology, for his valuable advice and suggestions as well as his continued guidance, patience and support that helped me to shape and refine my work.

My sincere thanks to **Dr. T. SREE SHARMILA**, Associate Professor, Department of Information Technology for her valuable suggestions throughout this project.

I express my deep respect to the founder **Dr. SHIV NADAR**, Chairman, SSN Institutions. I also express my appreciation to our **Dr. V. E. ANNAMALAI**, Principal, for all the help he has rendered during this course of study.

I would like to extend my sincere thanks to all the teaching and non-teaching staff of our department who have contributed directly and indirectly during the course of my project work. Finally, I would like to thank my parents and friends for their patience, cooperation and moral support throughout my life.

CHARULATHA S **DUVICKSHA U** **POOJA LAXMI S**

ABSTRACT

Evaluating answer scripts is always a strenuous task for teachers. And now in the era of online education and PDF answer scripts, this has become even more of an arduous work. This project aims to digitize answer scripts using semantic segmentation and aid professors in their evaluation. This consists of two parts; the first part is to extract the segments from the scanned answer script images while the second part is to perform OCR to convert the scanned images to text. Segment anything is a deep learning algorithm used for segmentation. Our system uses this model to segment the answers into 13 classes namely degree, diagram, heading, numbers, paragraphs, page, numbers, question number, register number, sign, subheading, subject name, table. The feature of automated mask generation was used which in turn segments the answers on its own understanding of the source image. The model was trained for our custom dataset which consisted of about 371 annotated images of answer scripts from the IT department. A model named docTR is used for Optical Character Recognition. This is a specialized type of OCR model that is trained to recognize and extract texts. Here, it is used to convert the handwritten text from the images into text that is easy for the system to read and evaluate the answers accordingly..

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Evaluation of answer scripts conducted in the traditional way is a time-consuming and a tedious task. Digitization involves the conversion of traditional paper-based answer scripts into digital formats, allowing them to be evaluated and stored electronically. This process involves the use of digital technologies such as scanning and optical character recognition (OCR). It can have numerous benefits for educational institutions, educators, and students. The use of digital technologies for evaluating answer scripts has become increasingly popular, with remote learning and evaluation becoming more common. This represents a significant shift towards a more efficient, accurate, and secure method of evaluation, and is likely to continue to gain importance in the education sector in the years to come.

1.2 NEED FOR STUDY

The COVID-19 pandemic has brought attention to the significance of digitising answer sheets. Numerous educational institutions have been forced to implement online learning and remote evaluation techniques as a result of the pandemic's disruption of conventional educational practises. To prevent physical contact and the propagation of the virus, digitizing answer scripts has become an important approach. Additionally, it enables remote work for evaluators, which is crucial during times of social isolation and lockdowns. Digital evaluations can also improve resource management for educational institutions by obviating the requirement for physical storage facilities.

Due to the system's many benefits, this trend of evaluation has persisted long after the pandemic has ended. Comparatively to the conventional way of assessing paper-based scripts, evaluators can quickly access and evaluate papers using digital answer scripts from any location, saving time and money. It can also guarantee reliable findings and lower the chance of errors brought on by human data entering. Students can track their progress over time and receive thorough feedback on their responses from digital assessments. Additionally, it offers a safe method of storing paper-based scripts, lowering the chance of loss or damage. Overall, digitizing answer scripts can streamline the evaluation process, increase efficiency, and provide valuable insights to students and educators.

1.3 OBJECTIVES

• To train a proposed deep-learning model to segment answers with respect to question paper structure.

• To segment the data contained in the answer scripts and include more number of segments in the pdf's.

• To train the model to segment various parts of answer such as each individual paragraphs, equations, tables, and diagrams.

• To make it easier for teachers to access the answers with the help of segments.

• To perform Optical character recognition on the processed image and choosing the best engine to our application on metrics such as accuracy, recall and precision.

CHAPTER 2

REVIEW OF LITERATURE

Segment Anything [1]

The Segment Anything project aims to bring foundation models and image segmentation together. The new task (promptable segmentation), model (SAM), and dataset (SA-1B) that enable this advancement are our main contributions. The model architecture is constrained by the promptable segmentation task and the aim of real-world application. To enable interactive use, the model must support flexible prompts, compute masks in amortised real-time, and be ambiguity-aware.Building a foundational model for image segmentation is the aim of this work.That is, they aim to create a model that is flexible and pre-trained on a large dataset using a task that allows for strong generalisation.With this model, they hope to use prompt engineering to address a variety of downstream segmentation issues on new data distributions. Task, model, and data are the three components that make this plan successful.

Segmenter: Transformer for Semantic Segmentation [5]

Segmenter, a transformer model for semantic segmentation, is introduced in this paper. In contrast to convolution-based techniques, the approach enables global context modelling at the network's first layer and throughout. In this paper, a pure transformer approach to semantic segmentation is introduced. The encoding portion builds on the recently released Vision Transformer (ViT), but is different in that we depend on all image patches being encoded. They note that the transformer does a great job of capturing the overall context. It contributes the following four things:(i) Novel method for semantic segmentation based on the Vision Transformer (ViT) that outperforms FCN-based approaches, does not rely on convolutions, and intentionally captures contextual information.(ii) A family of models is presented with various levels of resolution that enable a trade-off between accuracy and runtime, ranging from state-of-the-art performance to models with quick inference and good performances.

Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study [7]

The process of turning handwritten text from scanned or printed text images [1] into editable text for further processing is known as optical character recognition (OCR). The text can be automatically recognised by machines thanks to this technology. It resembles how the human body's eye and mind work together. Although the extracted text from the images can be read by the eye, the brain actually processes and interprets it. There may be a few issues with the computerised OCR system development. OCR software comes in a variety of forms today, including desktop, server, and web versions. Any OCR tool's accuracy rate can range from 71 to 98 percent. There are currently many OCR tools available, but very few of them are open source and free. One of the open source and free OCR programmes is called Tesseract. It is platform independent because it was written in C++. It can be accessed as a dynamic link library (DLL) and used in other programmes. To use the functionality offered by Tesseract, it can thus be simply added as a reference in the form of a DLL in other applications.

Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR) [8]

Optical Character Recognition (OCR) of documents has tremendous practical value given the prevalence of handwritten documents in human transactions. A science known as optical character recognition makes it possible to convert various kinds of documents or images into editable, searchable, and analyzable data. Researchers have been using artificial intelligence and machine learning tools to automatically analyse printed and handwritten documents over the past ten years in order to digitise them. This review paper's goals are to provide research directions and a summary of previous studies on character recognition in handwritten documents.Additionally, it was noted that Convolutional Neural Networks (CNN) are being used more frequently by researchers to recognise characters that have been manually and mechanically printed. This is because architectures based on CNN are well suited for recognition tasks with image input. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and other image object recognition tasks were among the first to use CNN [184]. The CNN-based architectures ResNet [187], GoogLeNet [186], and AlexNet [185] are some of the most popular ones for visual recognition tasks.

A Review on Deep Learning Techniques Applied to Semantic Segmentation [9]

Researchers working on computer vision and machine learning are becoming more and more interested in image semantic segmentation. Autonomous driving, indoor navigation, and even virtual or augmented reality systems, to name a few, require accurate and effective segmentation mechanisms. The rise of deep learning techniques in almost every area or application target related to computer vision, such as semantic segmentation or scene understanding, coincides with this demand. This article offers an overview of deep learning techniques for semantic segmentation used in a variety of application domains. The most reliable method, DeepLab, consistently outperforms the competition on nearly every RGB image dataset. Recurrent networks like LSTM-CF dominate 2.5D or multimodal datasets. PointNet has paved the way for future research on handling unordered point clouds without any kind of preprocessing or discretization, but 3D data segmentation still has a long way to go.

YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [10]

A novel real-time object detector architecture and associated model scaling strategy were proposed in this paper. Additionally, they discover that the process of object detection methods evolving produces new research topics. The replacement issue with the re-parameterized module and the allocation issue with the dynamic label assignment were discovered during the research process. The issue was addressed by the trainable bag-of-freebies method, which increases object detection accuracy. They created the YOLOv7 series of object detection systems, which produce cutting-edge results, based on the aforementioned information.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 GENERAL FLOW OF USE

The answer script is taken as a pdf format and split into images. Pre-processing the digital image can increase its quality by doing things like reducing noise, boosting contrast, or modifying brightness and colour. The annotated dataset is used to train an object detection model such as YOLOv7 or Segment Anything, which learns to detect the bounding boxes around the answers. It is fed to the Yolov7 for object detection and the Segment anything model for segmentation process. These segments are fed into the OCR model for converting the hand written text into text. The extracted data may be stored in a database or a file system for easy retrieval and analysis.

3.2 ML SERVICE

The ML models takes input a single image and returns the detected objects for the yolov7 model and semantic masks for

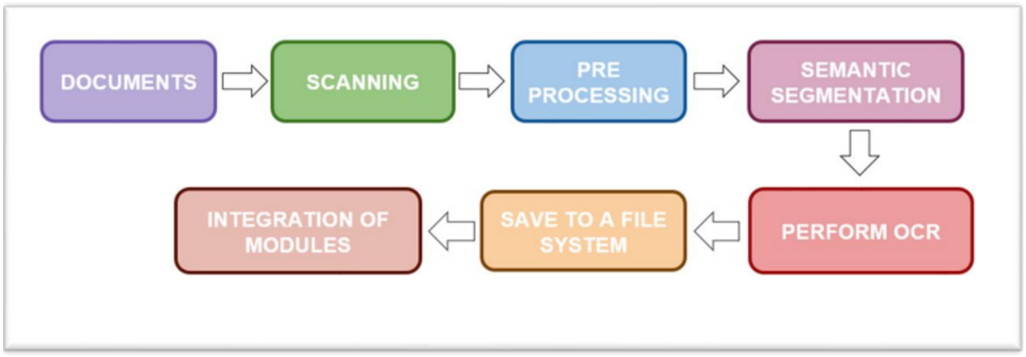


FIGURE 3.1: Workflow

segment anything model. For the ocr, the whole pdf is sent as an input. The ocr model creates bounding boxes for each word in the pdf and generates the corresponding text.

3.2.1 ENDPOINTS IN ML SERVICE

There are 3 endpoints in the ML Service:

- Yolov7: accepts an image and returns a image where the labels are detected
- SAM - stage1: accepts an image and returns an automatically generated semantic mask
- SAM - stage2: accepts an image and returns a segmented image and mask for the corresponding segment

3.2.2 MOTIVATION FOR RUNNING IT AS A SEPARATE APPLICATION

The Segment Anything model is run as a separate application to allow users to take advantage of the model's features on a variety of input data types and in various contexts. Users can label and extract particular regions of interest from images and videos by using the model as a standalone application, which eliminates the need for users to write custom code or integrate the model into their current applications. Running Segment Anything as a separate application also allows for greater flexibility and scalability, as the model can be deployed on different hardware and cloud infrastructure to accommodate varying workloads and performance requirements. The Deep Learning model used is a Segment Anything and requires GPU hardware for inference. Thus, we can deploy the model alone, and not the python script with GPU. This implementation also logically decouples the different modules for simpler handling.

3.2.3 MODEL

A deep learning-based segmentation technique called ”Segment Anything” can precisely recognise and separate out different objects in an image. The model is based on a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which work together to extract features from the input image and predict the boundaries of each object.

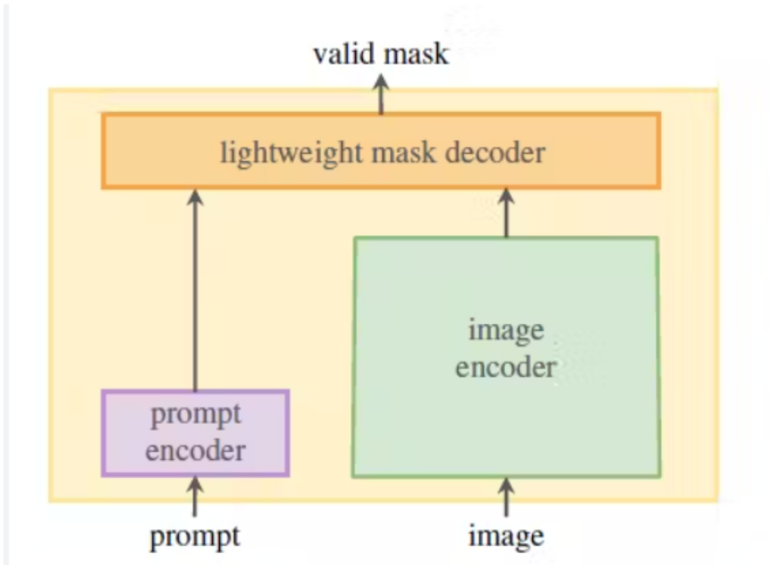


FIGURE 3.2: Segment Anything model architecture

The model works by first processing the input image through a series of convolutional layers, which extract low-level features such as edges and textures. These features are then fed into a set of encoder-decoder networks, which use a series of downsampling and upsampling operations to gradually increase the resolution of the feature maps while reducing the spatial dimension of the feature maps. The output of the encoder-decoder networks is a set of feature maps that encode information about the objects in the input image.

3.3 POST PROCESSING OF ML SERVICE RESPONSE

- Images, masks containing the necessary parts are retrieved from the segmented images.
- A pdf of the entire answer script is sent to the OCR model for converting handwriting to text
- An answer key needs to be created for each question in the exam. The answer key will contain the correct answer for each question, which will be used as a reference during evaluation.
- The evaluation software should be capable of automatically comparing the answers provided by the student to the answer key and providing a score for each question.
- The evaluation results need to be reviewed and validated to ensure accuracy. This can be done by human evaluators or through an automated validation process.
- The final scores for each student can be generated and stored in a digital format. The results can be viewed by students and faculty through a secure online portal.

.

3.4 OCR

Optical Character Recognition, is a technology that enables computers to recognize text characters within digital images or scanned documents. Typically, pattern recognition algorithms are used by OCR systems to recognise individual characters, which are then transformed into machine-readable text that can be searched, edited, or analysed.

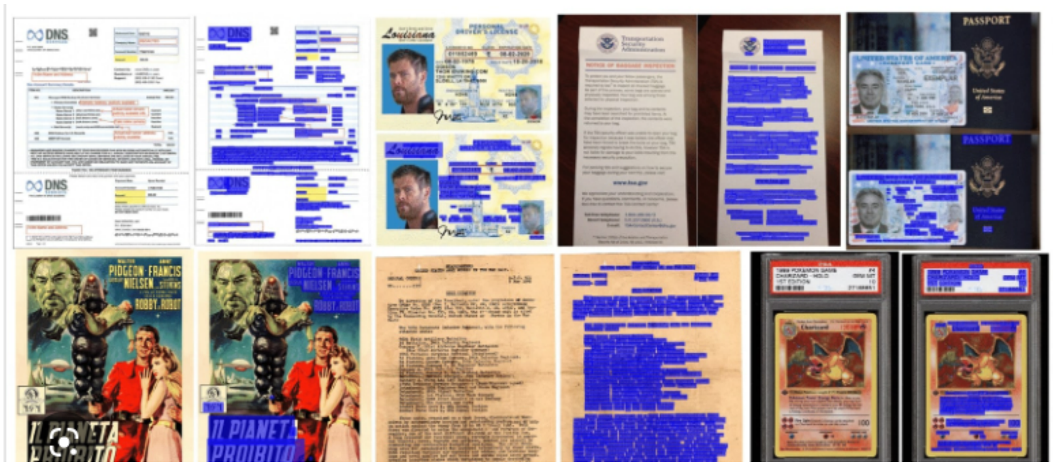


FIGURE 3.3: OCR on various documents

3.4.1 DOCTR-OCR

Doctr OCR is an open-source optical character recognition (OCR) software that is designed to help users extract text from various document types. It uses deep learning algorithms to recognize text and accurately extract it from scanned documents, images, and PDFs. It can handle a wide range of document types, including invoices, receipts, resumes, and other forms of structured and unstructured data. It supports multiple languages and can recognize a variety of fonts and character styles, making it a versatile OCR solution for organizations and individuals with diverse document processing needs.It is built on top of PyTorch, a popular deep learning framework, and leverages several pre-trained models to achieve high accuracy in text recognition.

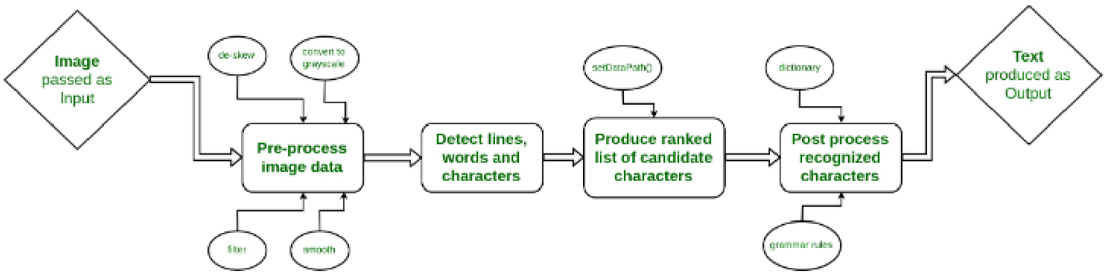


FIGURE 3.4: OCR architecture

The three main elements that make up Doctr OCR's core architecture are a document image reader, a text recognizer, and a post-processor. The document image reader reads a document or image as input and preprocesses it to improve the text's quality and readability. This could entail actions like noise reduction, contrast enhancement, and image scaling.

The text recognizer then receives the preprocessed image and applies a deep learning model to recognise and extract text from it. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs)-based models are among the pre-trained models that Doctr OCR supports. Users can, if necessary, fine-tune these models using their own data. Last but not least, a post-processor is used to carry out additional processing and cleanup tasks like spelling checks, language detection, and formatting on the extracted text. A number of formats, including plain text, JSON, and HTML, are available for saving the output.

CHAPTER 4

TOOLS AND TECHNOLOGIES USED

4.1 SOFTWARE REQUIREMENTS

- Python 3.8
- Pytorch 1.7
- Torchvision 0.8

4.2 HARDWARE REQUIREMENTS

- Minimum 16gb of RAM
- Minimum 16gb of VRAM (GPU Memory)

4.3 DEVELOPMENT ENVIRONMENT

- Platform Colab Notebook
- Operating System Windows

4.4 MODULES NEEDED

• OpenCV2

OpenCV is an open-source computer vision and machine learning software library that provides tools and algorithms for image and video processing, object detection and recognition, camera calibration, and more. It is extensively employed in numerous fields including robotics, augmented reality, and medical image analysis.

• OS

The Python os module gives you a means to use operating system-dependent functionality while interacting with the file system and operating system independent of any particular platform.

• SYS

SYS is another built-in module in Python. It offers access to some variables that the interpreter uses or maintains, as well as functions that have close relationships with the interpreter.

• Torch

Torch offers a strong foundation for creating and refining machine learning models. It is a Torch-based open-source machine learning library that is mostly used for deep learning applications. A wide range of features such as tensor computation, neural network building blocks, automatic differentiation, data loaders, and many more are offered by the torch module.

• JSON

Python's built-in Json module offers tools for interacting with JSON (JavaScript Object Notation) data. JSON is a simple data exchange format that is easy for both humans and robots to read, write, and generate.

CHAPTER 5

EXPERIMENTAL SETUP

The detection and segmentation of the answer scripts is experimented using two Machine learning algorithms. We use the YOLOv7 for detection of the classified labels and Segment Anything for segmentation of the answer scripts according to the given labels. Both the models were trained from a customized dataset of answer scripts. We used tensorflow, pytorch, numpy to develop and train the models. Both of these algorithms gave us positive results.

5.1 SEMANTIC SEGMENTATION OF ANSWER SHEETS

Assigning each pixel in an image to a particular semantic class, such as object, background, or region of interest, is a computer vision task known as semantic segmentation. Semantic segmentation offers a pixel-level understanding of the image content, in contrast to object detection, which only detects the presence of objects in an image. This is done by using a large annotated dataset to train a deep learning model, where each image has a segmentation mask labelled with the class that each pixel belongs to. Typically, the model is composed of an encoder-decoder architecture with skip connections, allowing the model to record both local and global context data.

5.1.1 MOTIVATION TO USE SEMANTIC SEGMENTATION

The necessity to extract precise and thorough information about an image's content drives the adoption of semantic segmentation. Semantic segmentation is a pixel-by-pixel classification task that gives each pixel in the image a label matching to the object or region it belongs to, unlike object detection, which just recognises the existence of items in an image. The main motivation for using semantic segmentation is to gain a better understanding of the content of an image, which can have many practical applications.

The use of semantic segmentation in evaluating answer scripts can help to automate the process of grading and provide more detailed feedback to students. The system can analyze each section separately and provide comments on the level of writing, the organization of the information, and the adherence to the assignment requirements by segmenting the answer script into distinct sections such as paragraphs, headings, and subheadings.

5.1.2 MOTIVATION FOR USING SEGMENT ANYTHING MODEL

The ”Segment Anything” model is a semantic segmentation model that uses deep learning techniques to accurately segment objects within an image. The model is based on the Fully Convolutional Network (FCN) architecture, which is designed to perform pixel-wise classification of images. The ”Segment Anything” model uses a series of convolutional layers to extract high-level features from the input image, followed by upsampling layers to generate a pixel-wise segmentation map. In order to combine data from various network levels and capture both low-level and high-level features of the image, the model uses skip connections. One of the key features of the ”Segment Anything” model is its ability to segment objects of different sizes and shapes, including irregular and complex shapes. The model accomplishes this by precisely segmenting each object in the image using a combination of local and global contextual information.

5.1.3 MOTIVATION FOR USING YOLOv7 MODEL

Modern object detection models like the YOLOv7 model have a number of advantages over older models. The high accuracy and speed of YOLOv7, which make it ideal for real-time applications like autonomous driving and video surveillance, are among the main reasons for using it. In contrast to models that use multiple stages or regions of interest, YOLOv7 accomplishes this by using a single deep neural network that predicts object bounding boxes and class probabilities simultaneously. Furthermore, YOLOv7 has a small memory footprint, making it possible for it to run on devices with limited resources like smartphones and drones. YOLOv7 is also highly customizable, making it appropriate for a variety of applications and domains.

5.2 ARCHITECTURE AND BACKGROUND

5.2.1 ARCHITECTURE AND BACKGROUND FOR SEGMENT ANYTHING MODEL

The Fully Convolutional Network (FCN) architecture, which is made for semantic image segmentation, is the foundation of the ”Segment Anything” model. An encoder-decoder network with skip connections makes up the architecture, which enables the model to record both low-level and high-level features of the image. The ”Segment Anything” model typically uses a pre-trained convolutional neural network (CNN) like VGG or ResNet as its encoder network. The encoder network is responsible for extracting high-level features from the input image. The encoder network is typically composed of convolutional layers that apply a series of filters to the input image to extract features at different scales.

A group of researchers from Facebook AI Research (FAIR) came up with the Segment-Anything model in 2021, which is a more recent computer vision model. The model is made to categorize each pixel in an image into a number of predefined categories or classes in order to perform semantic segmentation on the images.

We'll discuss the SAM (Segment Anything Model) for segmentation. SAM consists of three parts, as shown: an image encoder, a flexible prompt encoder, and a quick mask decoder. With specific trade-offs for (amortised) real-time performance, we build on Transformer vision models [22, 19, 23, 21]

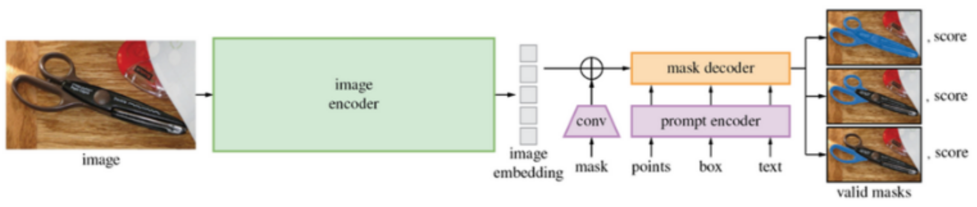


FIGURE 5.1: Overview of Segment Anything model

Image encoder: We use an MAE[17] pre-trained Vision Transformer (ViT)[19] minimally adapted to process high resolution inputs[21] because of its scalability and potent pre-training methods. Before prompting the model, the image encoder can be applied and executed once per image.

Prompt encoder: We take into account two sets of prompts: sparse (points, boxes, and text) and dense (masks).We used learned embeddings for each prompt type and free-form text with an off-the-shelf text encoder from CLIP[18] to represent points and boxes using positional encodings[20].

Convolutions are used to embed dense prompts (i.e., masks) and combine element-wise with the image embedding.

Maskdecoder: The image embedding, prompt embeddings, and an output token are efficiently mapped to a mask by a mask decoder. This design uses a modified Transformer decoder block (103), followed by a dynamic mask prediction head, and is inspired by [22, 23].To update all embeddings, our modified decoder block uses prompt self-attention and cross attention in two directions (prompt-to-image embedding and vice versa). After executing two blocks, we upsample the image embedding, and an MLP maps the output token to a dynamic linear classifier, which computes the mask for each image location's foreground probability.

5.2.2 ARCHITECTURE AND BACKGROUND FOR YOLOv7 MODEL

The YOLOv7 architecture is a deep neural network that consists of a backbone network and a detection head. The backbone network is typically a convolutional neural network (CNN) that extracts high-level features from the input image. In YOLOv7, the backbone network is a modified version of the popular EfficientNet architecture, which has been shown to achieve high accuracy with relatively few parameters.

Based on the features extracted by the backbone network, the detection head is in charge of forecasting object bounding boxes and class probabilities. The anchor boxes that define the sizes and aspect ratios of the objects to be detected are placed after a set of convolutional layers that make up the YOLOv7 detection head. A group of bounding boxes, each with a corresponding class probability, are the detection head's output.

The use of a ”SPP” (Spatial Pyramid Pooling) module, which enables the model to capture multi-scale features at various levels of the backbone network, is one of the key innovations of YOLOv7. Utilising ”attention” mechanisms to selectively weight the significance of various input image components during feature

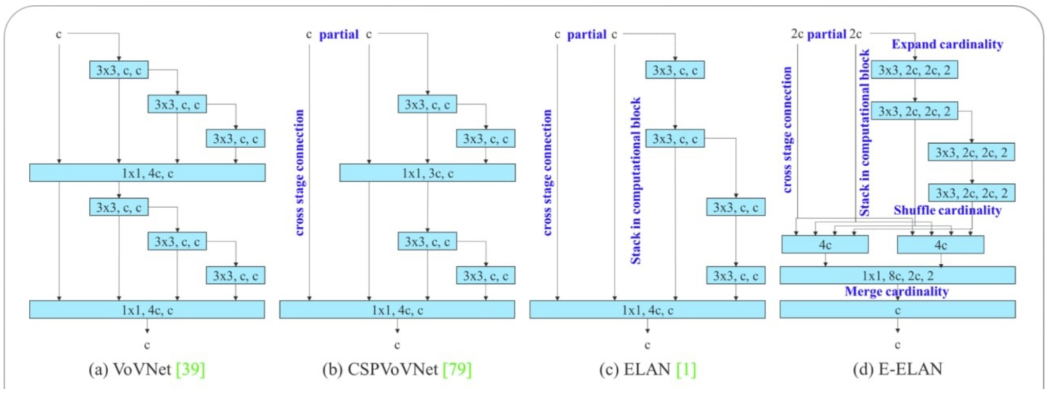


FIGURE 5.2: Architecture of yolov7

extraction is another crucial aspect of YOLOv7. This enables the model to ignore unimportant background noise and concentrate on important portions of the image.

5.3 DATASET USED

For the dataset, we have used the previous semester examination answer scripts of students.

The dataset consists of 370 images which has 13 labels

- Degree
- Diagram
- Heading

- Para
- Pgno
- Qno
- Regno
- Sign
- Subhead
- Subname
- Table
- Numbers
- Equation

5.3.1 DATA PREPARATION

We collected around 370 answer scripts that are representative of the actual semester answer sheets. In the next step, we annotated the answer scripts with bounding boxes around the corresponding labels marked for the entire dataset. The annotations should be accurate and consistent to ensure that the model is trained on high-quality data. For annotating images, we annotate the images using roboflow. After annotating we export the images in yolo v7 Pytorch and coco format.

5.3.2 TRAINING DATA

The Training data consists of 320 images that are manually annotated in roboflow and exported in coco format.

5.3.3 VALIDATION DATA

The validation data consists of 33 images.

5.3.4 TEST DATA

The Test data consists of 17 images.

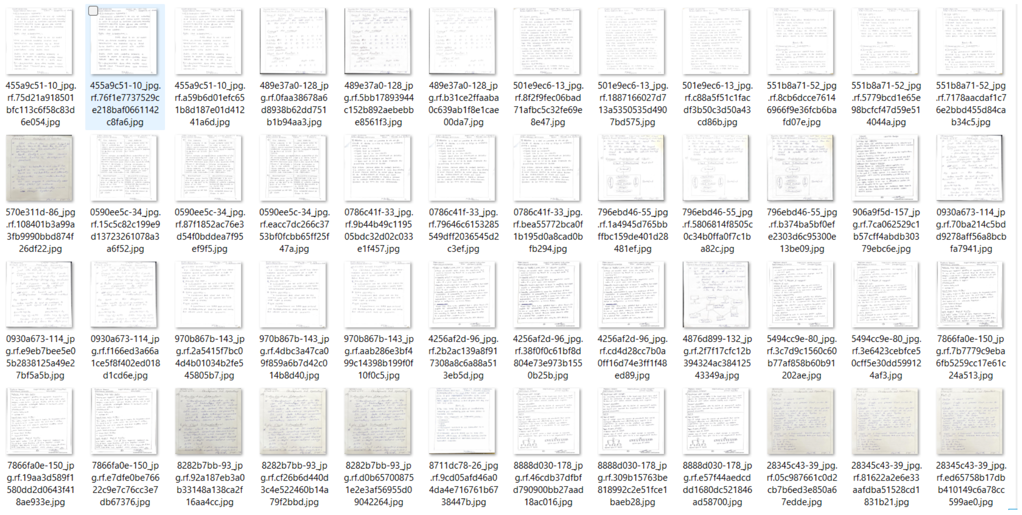
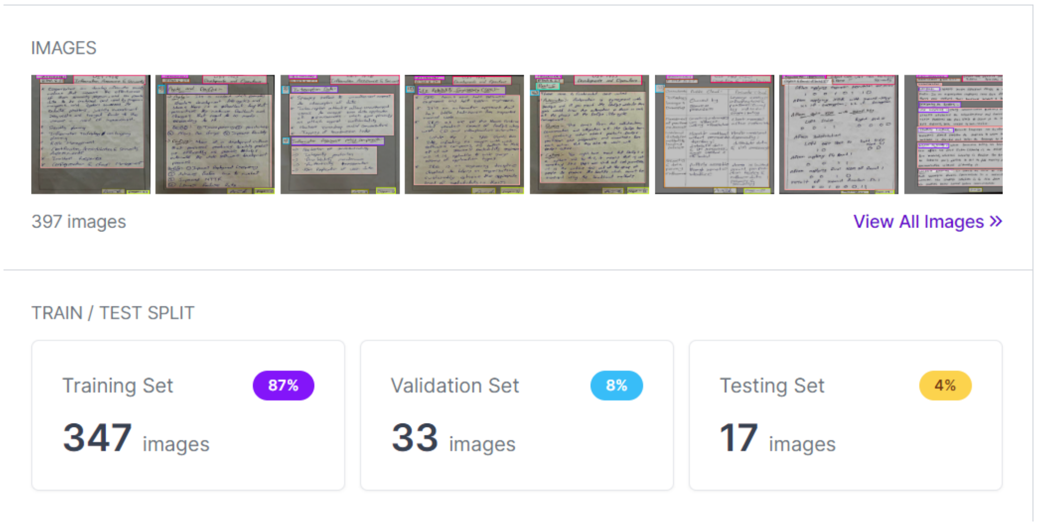


FIGURE 5.3: Dataset images



IMAGES

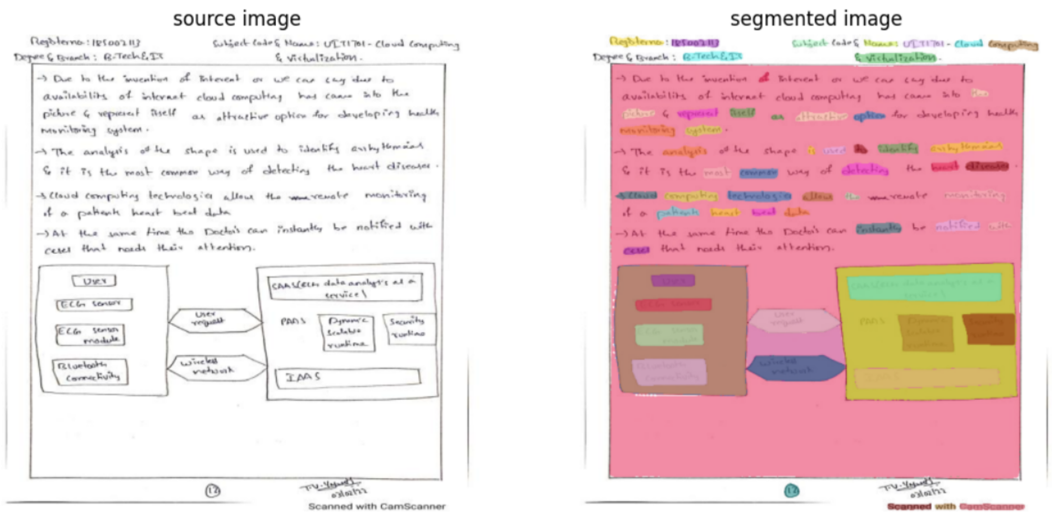
397 images

View All Images »

TRAIN / TEST SPLIT

Training Set 87%	Validation Set 8%	Testing Set 4%
347 images	33 images	17 images

FIGURE 5.4: Separation of the dataset images



Regd.no: 18500213   Subject Code & Name: UIT1701 - Cloud Computing & Virtualization.
Degree & Branch: B.Tech & IT

→ Due to the invention of Interent or we can say due to availability of internet cloud computing has came into the picture & represent itself as attractive option for developing health monitoring system.

→ The analysis of the shape is used to identify arrhythmias & it is the most common way of detecting the heart diseases.

→ Cloud computing technologies allow the ~~[illegible]~~ remote monitoring of a pattent heart beat data

→ At the same time the Doctor's can instantly be notified with cases that needs their attention.

FIGURE 5.5: Segment anything model automated mask generator

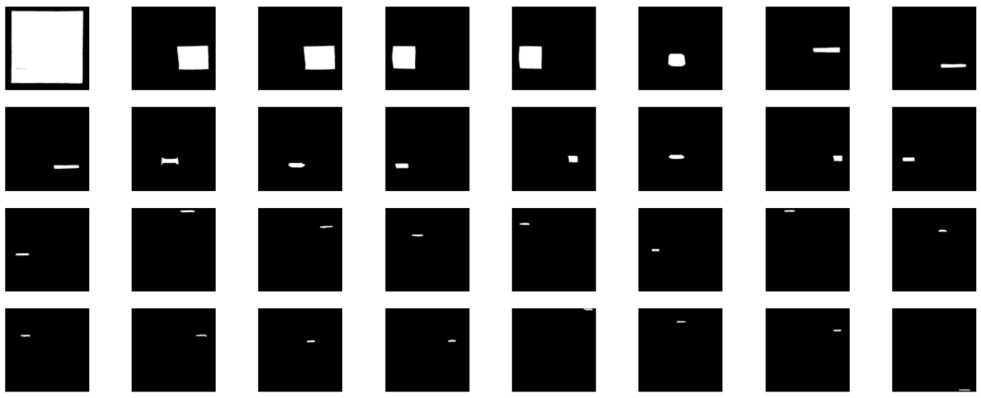


FIGURE 5.6: Mask mapping for the automated mask generator

5.4 TRAINING THE MODELS

5.4.1 TRAINING THE SEGMENT ANYTHING MODEL

The model is trained using a large dataset of labeled images, where each pixel in the image is labeled with the corresponding object or region it belongs to. During training, the model learns to associate each input pixel with its corresponding label, allowing it to accurately segment objects in new images. We trained the segment anything model over our dataset of 321 training images.

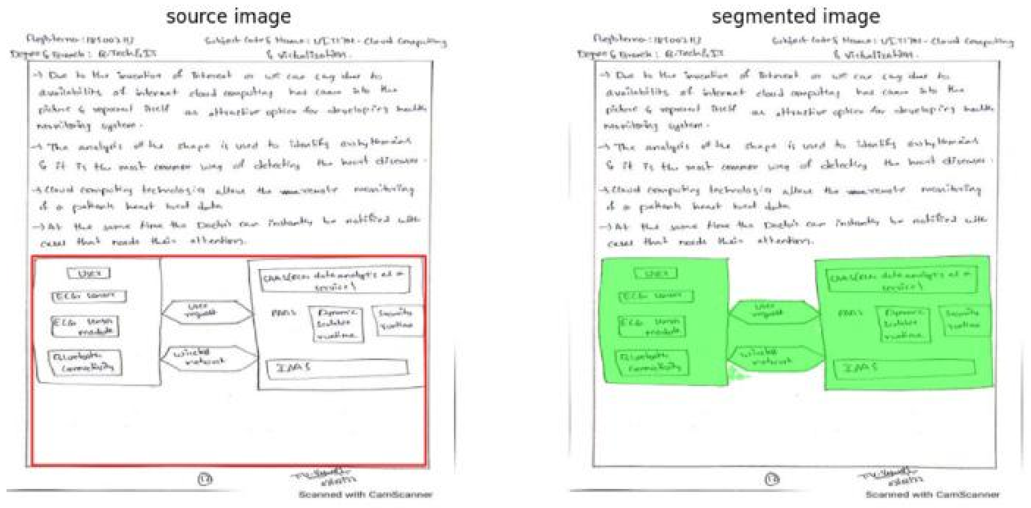


FIGURE 5.7: Bounded box and corresponding mask image

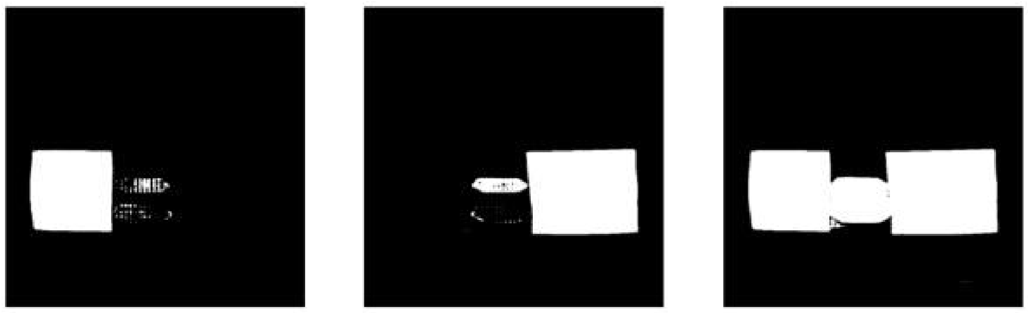
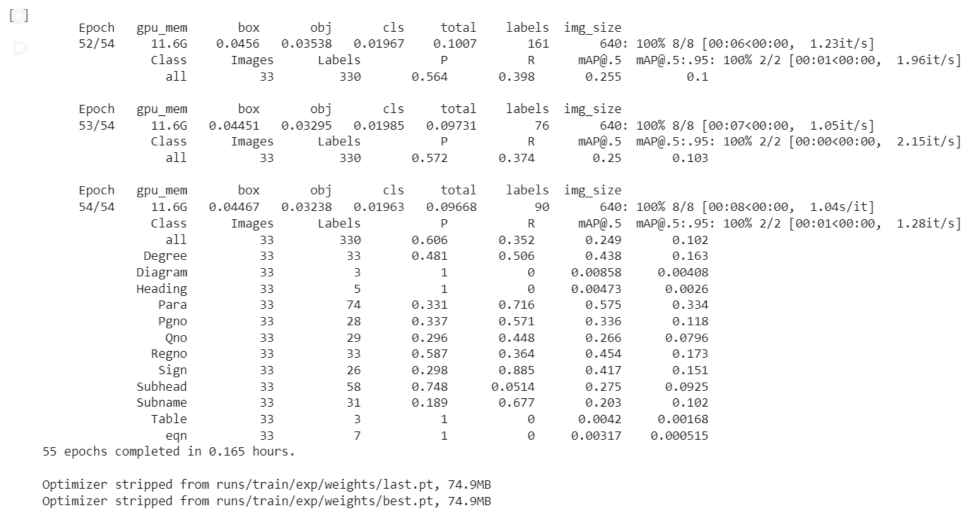


FIGURE 5.8: Mask mapping for the generated mask

5.4.2 TRAINING THE YOLOv7 MODEL

The model is trained on 370 images that are manually annotated in robolflow for the yolov7 pytorch format. It was trained for 55 epochs.



```
     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     52/54     11.6G    0.0456   0.03538   0.01967    0.1007       161       640: 100% 8/8 [00:06<00:00,  1.23it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 2/2 [00:01<00:00,  1.96it/s]
                 all          33         330       0.564       0.398       0.255         0.1

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     53/54     11.6G   0.04451   0.03295   0.01985   0.09731        76       640: 100% 8/8 [00:07<00:00,  1.05it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 2/2 [00:00<00:00,  2.15it/s]
                 all          33         330       0.572       0.374        0.25       0.103

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     54/54     11.6G   0.04467   0.03238   0.01963   0.09668        90       640: 100% 8/8 [00:08<00:00,  1.04s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 2/2 [00:01<00:00,  1.28it/s]
                 all          33         330       0.606       0.352       0.249       0.102
              Degree          33          33       0.481       0.506       0.438       0.163
             Diagram          33           3           1           0     0.00858     0.00408
             Heading          33           5           1           0     0.00473      0.0026
                Para          33          74       0.331       0.716       0.575       0.334
                Pgno          33          28       0.337       0.571       0.336       0.118
                 Qno          33          29       0.296       0.448       0.266      0.0796
               Regno          33          33       0.587       0.364       0.454       0.173
                Sign          33          26       0.298       0.885       0.417       0.151
             Subhead          33          58       0.748      0.0514       0.275      0.0925
             Subname          33          31       0.189       0.677       0.203       0.102
               Table          33           3           1           0      0.0042     0.00168
                 eqn          33           7           1           0     0.00317    0.000515
55 epochs completed in 0.165 hours.

Optimizer stripped from runs/train/exp/weights/last.pt, 74.9MB
Optimizer stripped from runs/train/exp/weights/best.pt, 74.9MB
```

FIGURE 5.9: Epochs run for the yolov7 model

CHAPTER 6

IMPLEMENTATION AND PSEUDO CODE

6.1 SEGMENTATION

6.1.1 INSTALLING THE MODEL

Once our environment is set up, download the SAM template into memory. When multiple states are available for inference, we can use the model in different ways to create masks. We investigate automated mask generation, generation of segmentation masks with jump boxes, and transformation of object detection dataset into segmentation masks. Three different encoders can be loaded into the SAM model: ViT-B, ViT-L and ViT-H. ViT-H is significantly better than ViT-B, but has only a marginal advantage over ViT-L. These encoders have a different number of parameters, ViT-B has 91M, ViT-L 308M and ViT-H 636M. This difference in size also affects the speed of reasoning, so it is necessary to choose the better encoder for our application.

6.1.2 AUTOMATED MASK GENERATION WITH SAM

To generate masks automatically we use the SamAutomaticMaskGenerator. This utility generates a list of dictionaries describing individual segmentations. Each dict in the result list has the following format:

• segmentation - [np.ndarray] - the mask with (W, H) shape, and bool type, where W and H are the width and height of the original image, respectively

• area - [int] - the area of the mask in pixels

• bbox - [List[int]] - the boundary box detection in xywh format

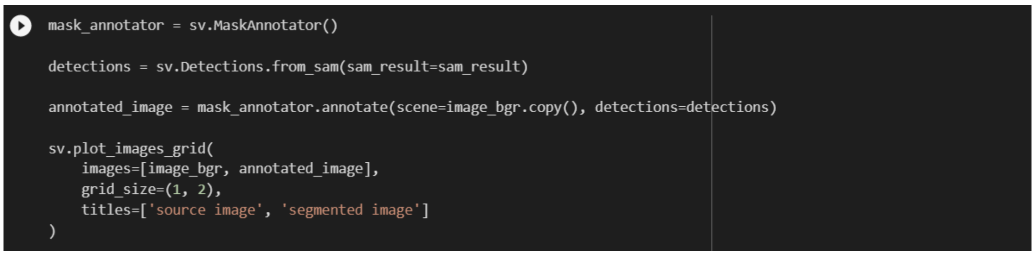
• predictediou - [float] - the model's own prediction for the quality of the mask

• pointcoords - [List[List[float]]] - the sampled input point that generated this mask

• stabilityscore - [float] - an additional measure of mask quality

• cropbox - List[int] - the crop of the image used to generate this mask in xywh format

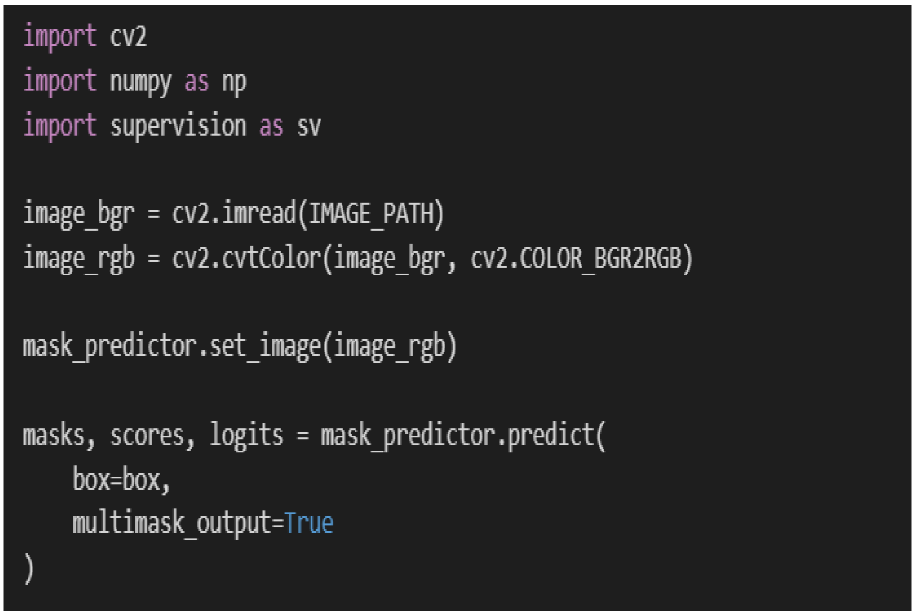
6.1.3 RESULT VISUALIZATION WITH SUPERVISION



```
mask_annotator = sv.MaskAnnotator()

detections = sv.Detections.from_sam(sam_result=sam_result)

annotated_image = mask_annotator.annotate(scene=image_bgr.copy(), detections=detections)

sv.plot_images_grid(
    images=[image_bgr, annotated_image],
    grid_size=(1, 2),
    titles=['source image', 'segmented image']
)
```

FIGURE 6.1: Visualization

Here we create an instance of the MaskAnnotator class from the supervision module and assign it to the mask_annotator variable. This class is responsible for drawing the segmentation mask on the input image. An instance of the Detections class from the supervision module is passed from the sam_result variable to its constructor and assigned to the detections variable. This class converts the binary segmentation mask returned by the ViT model to a format that can be used by the MaskAnnotator class. Then we call the annotate() method of the mask_annotator object with the original input image image_bgr and the detections variable as arguments. This method overlays the segmentation mask on the input image and returns the annotated image, which is assigned to the annotated_image variable. Finally we call the plot_images_grid() function from the supervision module to display a grid of images, which includes the original input image



```
import cv2
import numpy as np
import supervision as sv

image_bgr = cv2.imread(IMAGE_PATH)
image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)

mask_predictor.set_image(image_rgb)

masks, scores, logits = mask_predictor.predict(
    box=box,
    multimask_output=True
)
```

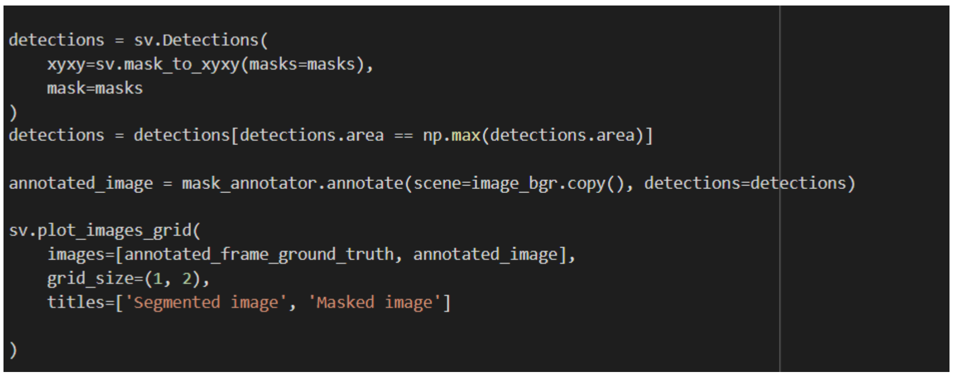
FIGURE 6.2: Mask Generation

(image_bgr) and the annotated image (annotated_image). The grid_size argument specifies the size of the grid, and the titles argument specifies the titles of each image.

6.1.4 GENERATE MASKS WITH SAM

Here the set_image method of the mask_predictor object is called with the image_rgb argument to set the image to be used for segmentation. The predict method of the mask_predictor object is called to perform segmentation on the image. The box argument specifies the bounding box of the object to segment. We set The multimask_output argument to True, to indicate that the method should return masks for all instances of the object in the image. The method returns three values: masks: A list of binary masks, one for each instance of the object in the image. scores: A list of confidence scores for each instance. logits: A list of logits for each instance.

6.1.5 BOUNDING BOX TO MASK



```
detections = sv.Detections(
    xyxy=sv.mask_to_xyxy(masks=masks),
    mask=masks
)
detections = detections[detections.area == np.max(detections.area)]

annotated_image = mask_annotator.annotate(scene=image_bgr.copy(), detections=detections)

sv.plot_images_grid(
    images=[annotated_frame_ground_truth, annotated_image],
    grid_size=(1, 2),
    titles=['Segmented image', 'Masked image']

)
```

FIGURE 6.3: Mask

The predicted masks are annotated in the image.The segmented image and masked image are plotted side by side using the plot_images_grid function from the supervision library. The snippet is shown in figure 6.3

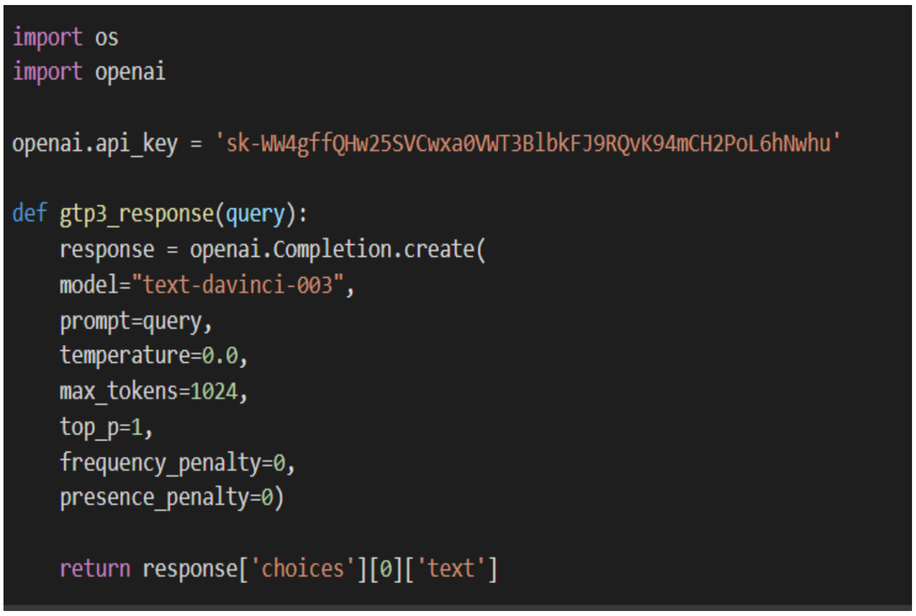
6.2 OCR

6.2.1 DETECTION AND RECOGNITION

Firstly, the image is pre-processed to improve the quality of text recognition. This can include operations such as resize, normalize, etc. Doctr OCR detects text areas using one of its pre-trained text recognizers. This can be done using several different methods such as CRAFT (Character Area Awareness for Text Detection) or EAST (Effective and Accurate Scene Text) detectors.Once the text areas are detected, text recognition is used to identify the text within them. Doctr OCR uses the open source OCR engine Tesseract OCR for text recognition. Recognized text can be stored in a text file or database.

6.2.2 TEXT GENERATION

The code defines a function called gtp3_response that utilizes the OpenAI API to generate a response to a given query using the GPT-3 language model. The function takes a single argument query, which is the input prompt for the language model.



```python
import os
import openai

openai.api_key = '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'

def gtp3_response(query):
    response = openai.Completion.create(
    model="text-davinci-003",
    prompt=query,
    temperature=0.0,
    max_tokens=1024,
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0)

    return response['choices'][0]['text']
```

FIGURE 6.4: Generating text

The function makes use of the openai.Completion.create method to generate a response using the text-davinci-003 model. It sets the temperature parameter to 0.0, which disables random sampling and ensures that the model generates the most probable token at each step. The max_tokens parameter sets the maximum number of tokens the model can generate, and top_p sets the cumulative probability threshold for the generated tokens. The frequency_penalty and presence_penalty parameters control the likelihood of the model repeating or omitting phrases from the input prompt.

Finally, the function returns the generated text by accessing the text field of the first element in the choices list of the response object.

CHAPTER 7

PERFORMANCE ANALYSIS

7.1 PERFORMANCE ANALYSIS FOR SEMANTIC SEGMENTATION

This section explains the pertinent metrics that were used to evaluate the effectiveness of the semantic segmentation.

7.1.1 F1 SCORE

A binary classification model's performance is assessed using the F1 score metric. It combines recall and precision to produce a single score that represents the model's overall accuracy.

The F1 score is determined by dividing the number of true positives by the total number of true positives and false positives, and the number of true positives by the total number of true positives and false negatives, respectively.

The F1 score formula is:

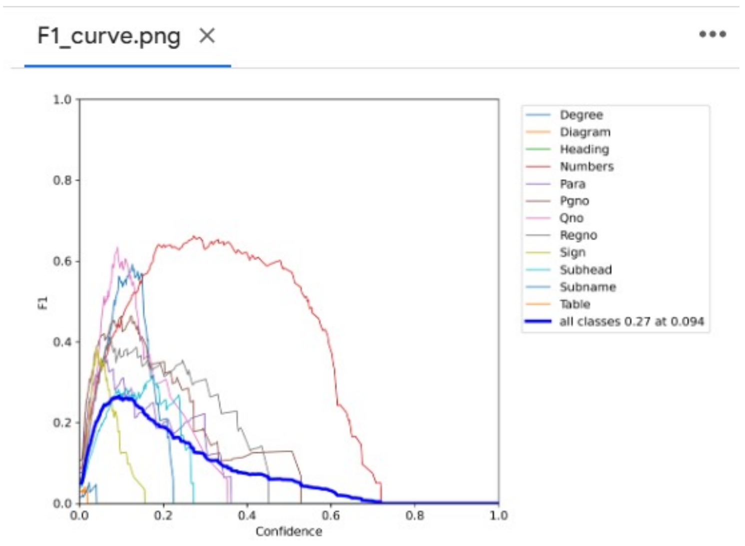


FIGURE 7.1: F1 Curve

F1 score = 2 * (precision * recall) / (precision + recall)

The F1 score has a range of 0 to 1, with a score of 1 denoting perfect precision and recall and a score of 0 denoting subpar performance.

7.1.2 PR CURVE

A binary classification model's performance is graphically represented by a precision-recall (PR) curve, which is especially useful when the class distribution is unbalanced. It shows the relationship between recall (x-axis) and precision (y-axis) for various threshold values that the model uses to make predictions.

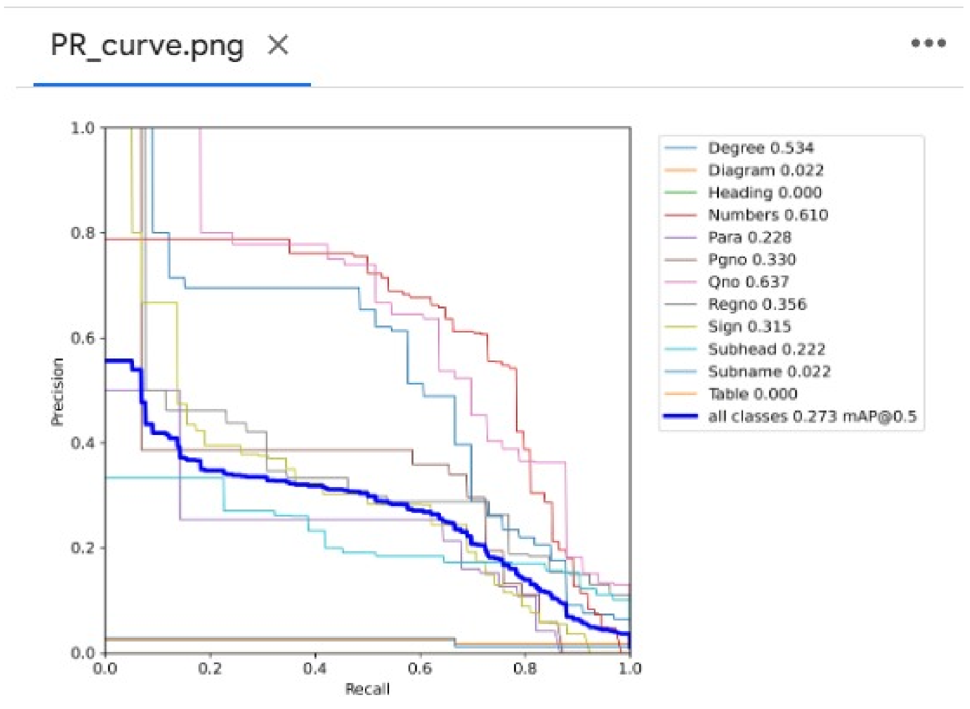


FIGURE 7.2: PR Curve

Precision is the percentage of true positives among all positive instances while precision is the percentage of true positives (TP) among all instances predicted as positive (TP + false positives or FP). The PR curve demonstrates how precision and recall change as the model's prediction threshold changes. In contrast to ROC curves, which plot true positive rate against false positive rate, PR curves provide a better visualisation of the trade-off between precision and recall when evaluating models on imbalanced datasets.

7.1.3 R CURVE

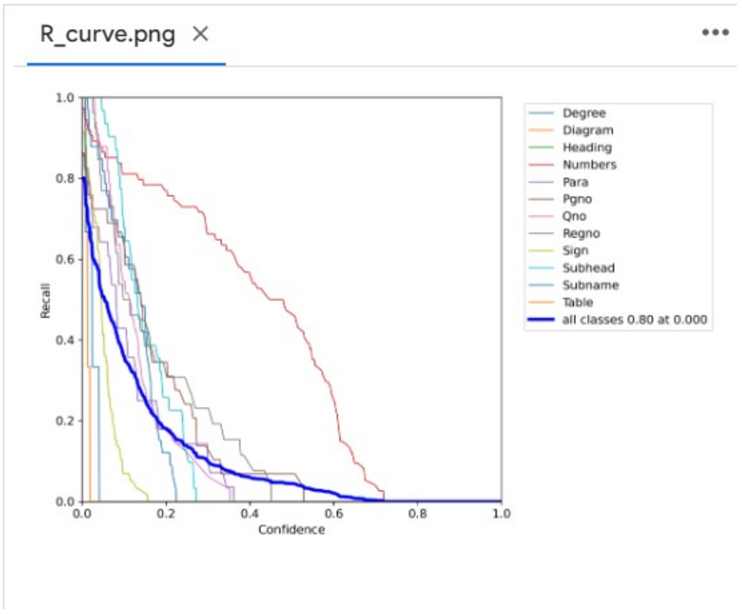


FIGURE 7.3: R Curve

For a specific classification task, a recall curve is a graph that depicts the relationship between the recall (true positive rate) and the quantity of retrieved instances. It is frequently used in machine learning and information retrieval to assess how well retrieval systems like search engines, recommender systems, and object detectors are performing. The recall curve is produced by progressively increasing the number of retrieved instances while continuously measuring recall. Normally, the curve begins with zero recall and rises as more pertinent instances are retrieved. The curve may reach a point where it reaches a plateau, meaning that the majority of the important instances have already been retrieved. The curve's shape can reveal information about the retrieval system's effectiveness, including the efficiency of the ranking algorithm or the calibre of the features that are used for retrieval.It is possible to determine the system's strengths and weaknesses and make improvements by looking at the curve's shape.

7.1.4 CONFUSION MATRIX

The effectiveness of a classification model is assessed using a table called a confusion matrix.

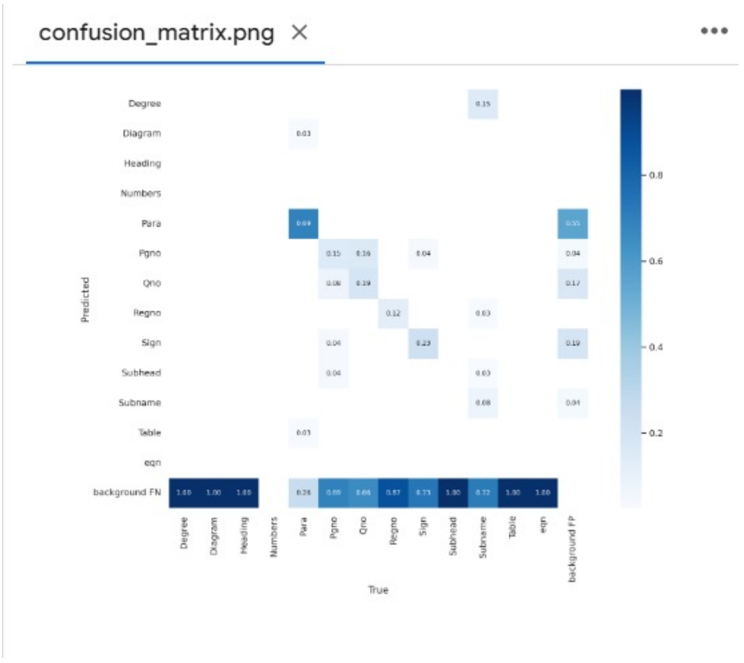


FIGURE 7.4: Confusion matrix

In addition to comparing the predicted classes to the actual classes, it displays the proportion of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).The confusion matrix gives a more thorough understanding of the model's performance than just accuracy, making it a useful tool for assessing the effectiveness of classification models. Metrics like precision, recall, F1 score, and accuracy can be computed using it, among other things. The confusion matrix is shown in figure 7.4.

7.1.5 CLASSIFICATION AND PRECISION TABLE

Figure 7.5 is an illustration of the segmentation's outcome.

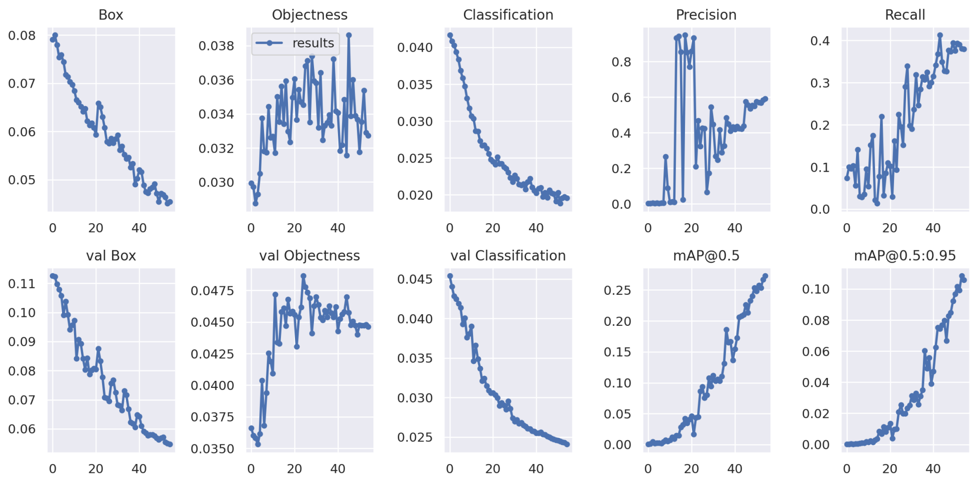


FIGURE 7.5: Result of Semantic Segmentation

CHAPTER 8

RESULTS

8.1 SEMANTIC SEGMENTATION USING SEGMENT ANYTHING MODEL

8.1.1 RESULTS VISUALIZATION WITH SUPERVISION

Figure 8.1 depicts the automatically produced masks created with the Segment Anything model using the Sam Automatic Mask Generator. With the help of this utility, a list of dictionaries describing individual segmentation is produced.

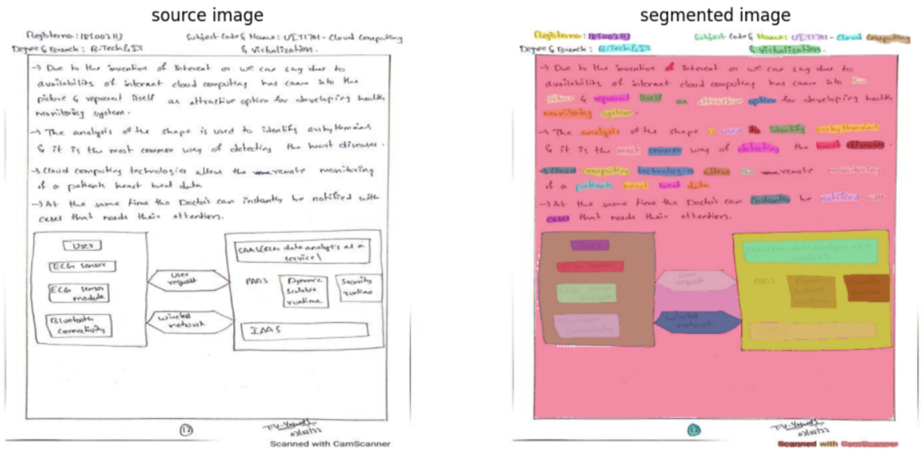


FIGURE 8.1: Visualisation of source image and automatically generated segmentation masks

8.1.2 SEGMENTED MASKS

Each segmentation that was obtained is highlighted separately in figure 8.2.

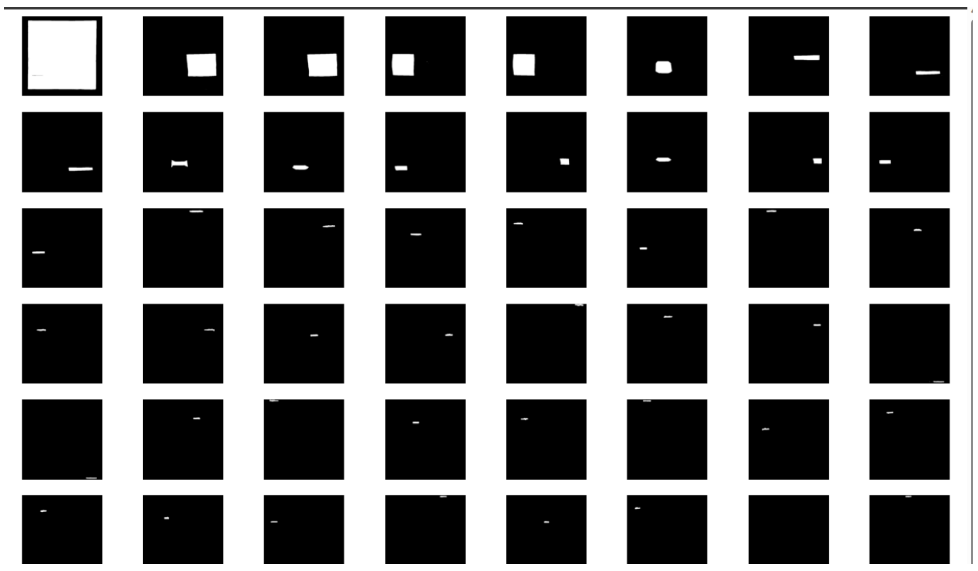


FIGURE 8.2: Segmented masks

8.1.3 SEGMENTED MASK WITH BOUNDING BOX

The output format of the automated mask generator differs from that of the mask predictor. The SAM model uses a bounding box format of [xmin, ymin, xmax, ymax]np.array.

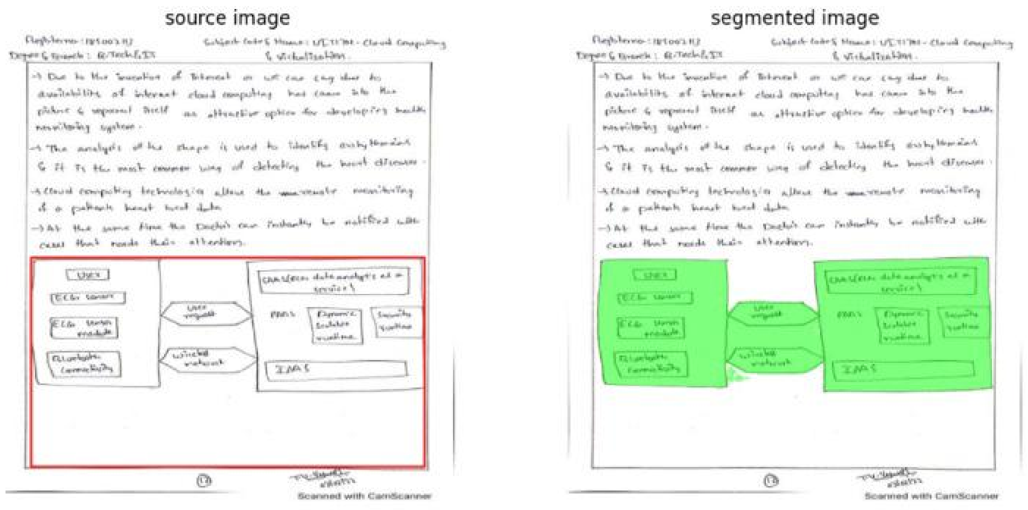


FIGURE 8.3: Segmented mask with bounding box

8.1.4 SEGMENTED MASKS GENERATED BY MASK PREDICTOR

Figure 8.5 illustrates the segmented masks generated by the mask predictor.

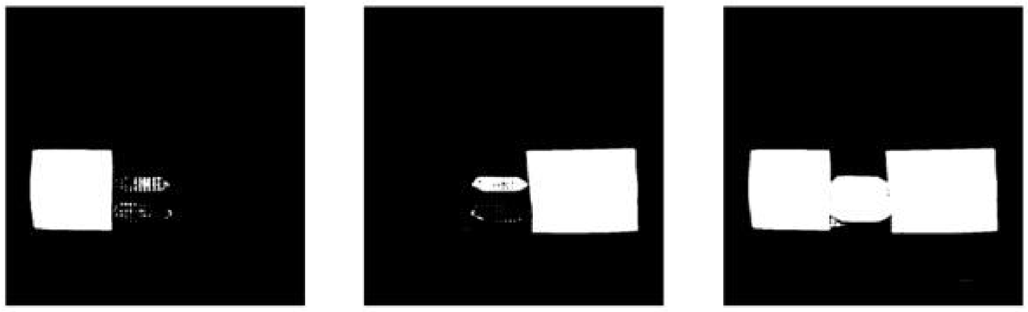
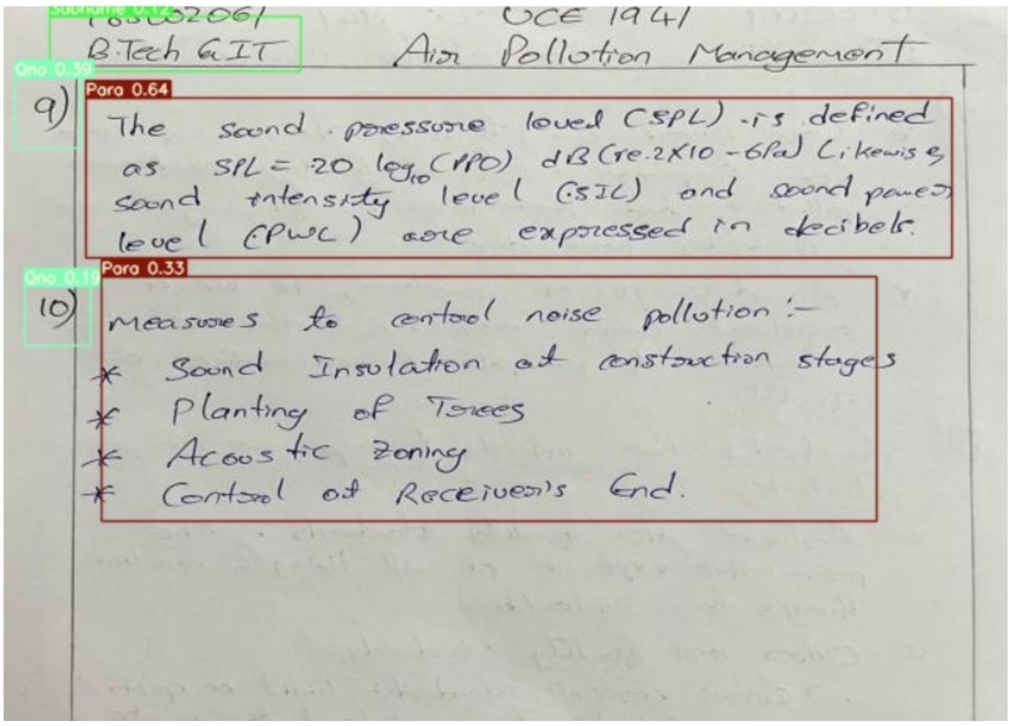


FIGURE 8.4: Segmented mask

8.2 SEGMENTATION USING YOLOV7 - PYTORCH

Figure 8.6 illustrates the segmentation done by YOLOV7 pytorch model.



Subname 0.72
Qno 0.39
Para 0.64
Qno 0.19
Para 0.33

OCE 1941
B.Tech GIT Air Pollution Management

9) The sound pressure level (SPL) is defined as $SPL = 20 \log_{10} (P/P_0)$ dB (re. 2X10 -6Pa) Likewise sound intensity level (SIL) and sound power level (PWL) are expressed in decibels.

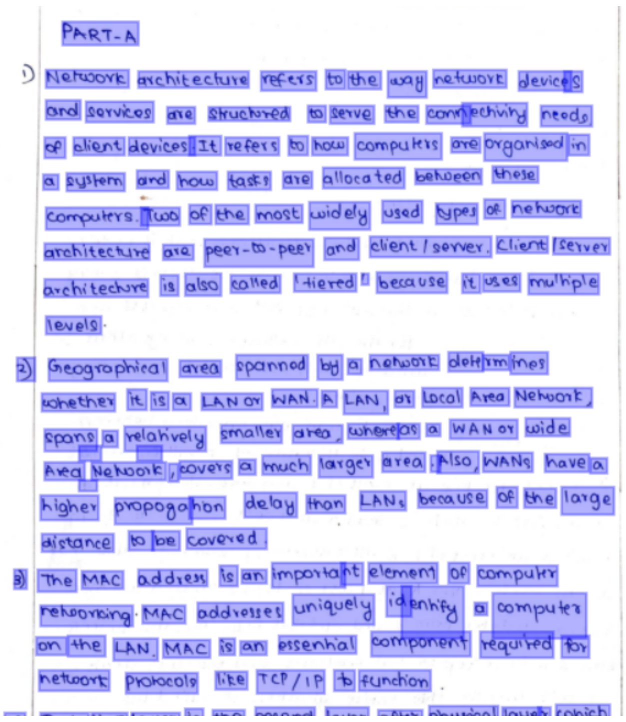
10) Measures to control noise pollution :-
* Sound Insulation at construction stages
* Planting of Trees
* Acoustic Zoning
* Control at Receiver's End.

FIGURE 8.5: Segmented answer scripts using Yolov7

8.3 OPTICALCHARACTER RECOGNITION

8.3.1 TEXT RECOGNITION USING DOCTR OCR

In docTR, end-to-end OCR is accomplished by first performing text detection (localising words), followed by text recognition (identifying all characters in the word). Figure 8.7 depicts the text recognised by the OCR model.



PART-A

1) Network architecture refers to the way network devices and services are structured to serve the connecting needs of client devices. It refers to how computers are organised in a system and how tasks are allocated between these computers. Two of the most widely used types of network architecture are peer-to-peer and client / server. Client / Server architecture is also called 'tiered' because it uses multiple levels.

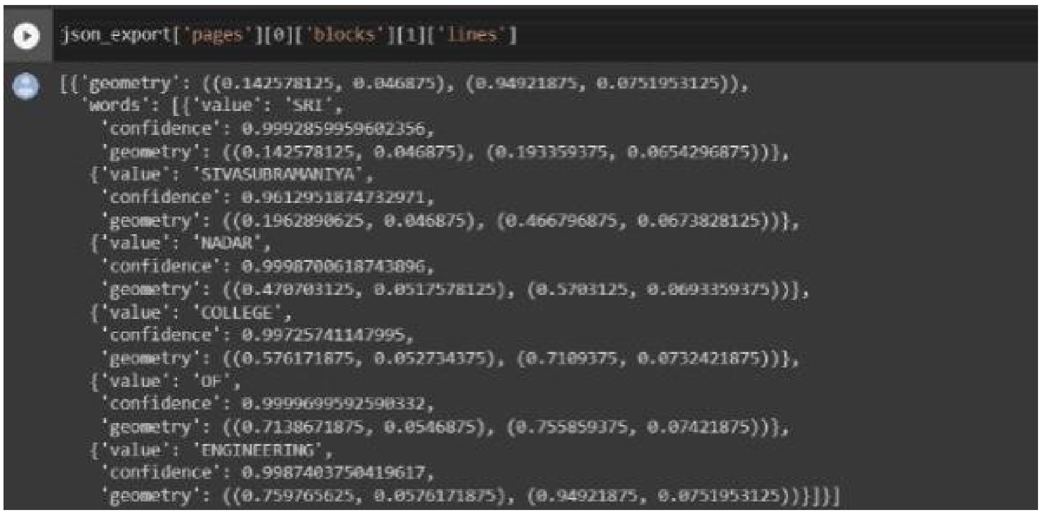
2) Geographical area spanned by a network determines whether it is a LAN or WAN. A LAN, or Local Area Network, spans a relatively smaller area, whereas a WAN or wide Area Network, covers a much larger area. Also, WANs have a higher propogation delay than LANs because of the large distance to be covered.

3) The MAC address is an important element of computer networking. MAC addresses uniquely identify a computer on the LAN. MAC is an essential component required for network protocols like TCP / IP to function.

FIGURE 8.6: OCR blocks

8.3.2 JSON REPRESENTATION OF THE RECOGNISED TEXTS

The Document object that the ocr predictor returns has a nested structure with Page, Block, Line, Word and Artefact. Additionally, it can also be exported in a better suited JSON format.



```
json_export['pages'][0]['blocks'][1]['lines']
```

```
[{'geometry': ((0.142578125, 0.046875), (0.94921875, 0.0751953125)),
  'words': [{'value': 'SRI',
    'confidence': 0.9992859959602356,
    'geometry': ((0.142578125, 0.046875), (0.193359375, 0.0654296875))},
   {'value': 'SIVASUBRAMANIYA',
    'confidence': 0.9612951874732971,
    'geometry': ((0.1962890625, 0.046875), (0.466796875, 0.0673828125))},
   {'value': 'NADAR',
    'confidence': 0.9998700618743896,
    'geometry': ((0.470703125, 0.0517578125), (0.5703125, 0.0693359375))},
   {'value': 'COLLEGE',
    'confidence': 0.99725741147995,
    'geometry': ((0.576171875, 0.052734375), (0.7109375, 0.0732421875))},
   {'value': 'OF',
    'confidence': 0.9999699592590332,
    'geometry': ((0.7138671875, 0.0546875), (0.755859375, 0.07421875))},
   {'value': 'ENGINEERING',
    'confidence': 0.9987403750419617,
    'geometry': ((0.759765625, 0.0576171875), (0.94921875, 0.0751953125))}]}]
```

FIGURE 8.7: JSON representation

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

9.1 CONCLUSION

In conclusion, our project is very practical, convenient, and has a wide application in the field of education. When done the old-fashioned way, evaluating answer scripts is a laborious and time-consuming task. To get around this, we have created a system that aims to digitise scripts for answers using Deep Learning methodologies and store them as responses to various questions. With the aid of segments, it makes it simpler for teachers to access the solutions.

9.1.1 BENEFITS

- The project has a lot of opportunity to conduct centralized/online tests and evaluations.
- One of the most crucial steps in any examination process is the evaluation of the answer sheets.

- The manual review of the answer sheets can occasionally be time-consuming and problematic in terms of security.
- The evaluation process is simplified and accelerated by using digital answer sheet evaluation.

9.1.2 FUTURE SCOPE OF DIGITISATION

Large-scale digitization projects may require specialized equipment, software, and personnel, while smaller projects may be performed using off-the-shelf tools and techniques. However, regardless of the size and scope of the project, the digitization of answer sheets can provide significant benefits such as improved accuracy, efficiency, and accessibility of the data.

9.2 FUTURE WORK

The development of more sophisticated techniques and tools for data capture, processing, and analysis is a part of the future work of digitising answer sheets. The automation of data extraction and verification processes through the use of artificial intelligence (AI) and machine learning (ML) algorithms is one example of this.

Another is the creation of cloud-based platforms for the management and archiving of digital data.

1) Increase the accuracy for Optical Character Recognition: OCR accuracy can be increased by using pre-processing methods like noise reduction, image binarization, skew correction, and contrast enhancement.By using dictionaries and word lists. OCR engines can be trained to recognise particular words and phrases. For particular document types, accuracy may be improved by doing this. The engine can recognise a wide range of text patterns by training them with a large dataset of sample images.

2) Integrate into a web application: An evaluation can be done by the evaluator using a web application while they are logged in from an isolated location. A marksheet can be automatically generated by the web application with some programming. We could create a query system that would allow us to display the answers in a more flexible way rather than showing all of a student's responses.

Ultimately, digitising answer sheets holds great promise for enhancing the effectiveness, efficiency, and efficiency of educational systems as well as for expanding the field of education more broadly.

CHAPTER 10

REFERENCES

[1] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick "Segment Anything",2023.

[2] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, Rohit Girdhar " Masked-attention Mask Transformer for Universal Image Segmentation",2022.

[3] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, Hengshuang Zhao "FocalClick: Towards Practical Interactive Image Segmentation", 2022

[4] Enze Xie1 , Wenhai Wang2 , Zhiding Yu3 , Anima Anandkumar3,4 , Jose M. Alvarez3 , Ping Luo1 "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers" Neural Information Processing Systems (NeurIPS 2021).

[5] Robin Strudel, Ricardo Garcia, Ivan Laptev, Cordelia Schmid "Segmenter: Transformer for Semantic Segmentation" ICCV 2021.

[6] Karez Abdulwahhab Hamad, Mehmet Kaya et al. “A Detailed Analysis of Optical Character Recognition Technology”, 2016.

[7] Chirag Indravadanbhai Patel, Dharmendra Patel, Atul Patel et al. “Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study”, 2012.

[8] Jamshed Memon, Maira Sami, Rizwan Ahmed KhaN et al. “Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)”, 2020.

[9] A. Garcia-Garcia, S. Orts-Escolano, S.O. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez et al, “A Review on Deep Learning Techniques Applied to Semantic Segmentation”, 2017.

[10] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese et al. “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression”, Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[11] R. Zhao et al., ”Rethinking Dice Loss for Medical Image Segmentation,” 2020 IEEE International Conference on Data Mining (ICDM), 2020, pp. 851-860, doi: 10.1109/ICDM50108.2020.00094.

[12]Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dolla´r (Feb 2015) Microsoft COCO: Common Objects in Context Microsoft Research, arXiv:1405.0312.

[13] A. Chaurasia and E. Culurciello, ”LinkNet: Exploiting encoder representations for efficient semantic segmentation,” 2017 IEEE Visual Communications and Image Processing (VCIP), 2017, pp. 1-4, doi: 10.1109/VCIP.2017.8305148.

[14] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao et al, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”, 2022.

[15]Guo, Y., Liu, Y., Georgiou, T. et al. “A review of semantic segmentation using deep neural networks”, 2018.

[16] L. -C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille et al. ”DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. [17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li,Piotr Doll´ar, RossGirshick et al.

"Masked Autoencoders Are Scalable Vision Learners" CVPR , 2022.

[18]]Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning Transferable Visual models from natural language supervision" ICML, 2021.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An image is worth 16x16 words: Transformers for image recognition at scale", ICLR,2021.

[20] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. "Fourier features let networks learn high frequency functions in low dimensional domains" NeurIPS, 2020

[21] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. "Exploring plain vision transformer backbones for object detection" ECCV, 2022

[22] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. “End-to-end object detection with Transformers” ECCV, 2020.

[23] Bowen Cheng, Alex Schwing, and Alexander Kirillov. “Per pixel classification is not all you need for semantic segmentation” NeurIPS, 2021.

[24] O. Ronneberger, P. Fischer and T. Brox et al. ”U-net: Convolutional networks for biomedical image segmentation”, International Conference on Medical image computing and computer-assisted intervention, 2015.

[25] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. Zisserman, A et al. “The PASCAL Visual Object Classes Challenge”, 2012.

[26] He, Kaiming Zhang, Xiangyu Ren, Shaoqing Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90.

[27] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba et al. “Semantic understanding of scenes through the ADE20K dataset”, 2016.

[28]Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, C. Lee Giles et al “Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks”, 2017.

[29] P. Y. Simard, D. Steinkraus and J. C. Platt et al, ”Best practices for convolutional neural networks applied to visual document analysis,” Seventh International Conference on Document Analysis and Recognition, 2003.

[30] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia et al, ”Pyramid Scene Parsing Network,” IEEE Conference on Computer Vision and Pattern Recognition 2017.

[31] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, et al. “Semantic understanding of scenes through the ADE20K dataset”, 2016.

[32] A. Bagdanov and J. Kanai et al, ”Projection profile based skew estimation algorithm for JBIG compressed images,” Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1997.