NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

POSTGRADUATE PROGRAM IN "DATA SCIENCE AND INFORMATION TECHNOLOGIES"
Direction: Big Data and Artificial Intelligence

MSc THESIS

Investigating Neural Networks and Transformer Models for Enhanced Comic Decoding

Eleni Ioanna P. Kouletou

Supervisors: Vassilis Papavasileiou, Associate Researcher ILSP/Athena R.C.
Vassilis Katsouros, Research Director ILSP/Athena R.C.

ATHENS

JULY 2024

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ "ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ ΠΛΗΡΟΦΟΡΙΑΣ"
Κατεύθυνση: Μεγάλα Δεδομένα και Τεχνητή Νοημοσύνη

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διερεύνηση μοντέλων νευρωνικών δικτύων και transformers για βελτιωμένη αποκωδικοποίηση κόμικ

Ελένη Ιωάννα Π. Κουλέτου

Επιβλέποντες: Βασίλης Παπαβασιλείου, Συνεργαζόμενος Ερευνητής ΙΕΛ/ΕΚ Αθηνά
Βασίλης Κατσούρος, Ερευνητής Α' ΙΕΛ/ΕΚ Αθηνά

ΑΘΗΝΑ

ΙΟΥΛΙΟΣ 2024

MSc THESIS

Investigating Neural Networks and Transformer Models for Enhanced Comic Decoding

Eleni Ioanna P. Kouletou
S.N.: 7115152100007

SUPERVISORS: **Vassilis Papavasileiou**, Associate Researcher ILSP/Athena R.C.
Vassilis Katsouros, Research Director ILSP/Athena R.C.

EXAMINATION COMMITTEE: **Vassilis Papavasileiou**, Associate Researcher ILSP/Athena R.C.
Vassilis Katsouros, Research Director ILSP/Athena R.C.
Vassilis Gatos, IIT/RS Demokritos NKUA

Examination Date: 8 July, 2024

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διερεύνηση μοντέλων νευρωνικών δικτύων και transformers για βελτιωμένη αποκωδικοποίηση κόμικ

Ελένη Ιωάννα Π. Κουλέτου
Α.Μ.: 7115152100007

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Βασίλης Παπαβασιλείου**, Συνεργαζόμενος Ερευνητής ΙΕΛ/ΕΚ Αθηνά
Βασίλης Κατσούρος, Ερευνητής Α' ΙΕΛ/ΕΚ Αθηνά

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Βασίλης Παπαβασιλείου**, Συνεργαζόμενος Ερευνητής ΙΕΛ/ΕΚ Αθηνά
Βασίλης Κατσούρος, Ερευνητής Α' ΙΕΛ/ΕΚ Αθηνά
Βασίλης Γάτος, Ερευνητής Α' ΙΠ&Τ/ΕΚ Δημόκριτος

Ημερομηνία Εξέτασης: 8 Ιουλίου 2024

ABSTRACT

Comic books, merging art with narrative, continue to captivate readers, cinema producers, and collectors, maintaining their allure as a cherished form of visual storytelling across decades. Comic image segmentation is a pivotal aspect of the digital transformation of comics. Leveraging heuristic approaches, neural network-based models (YOLO), and innovative transformer-based architectures (GroundingDINO, SAM), our research aims to autonomously segment comic pages into their fundamental components: panels, comic characters, and text areas. To this end, we further trained YOLOv5 and YOLOv8 models to identify these components, while transformer-based models employed prompts to retrieve them. By comparing their performance, in terms of established metrics (Precision, Recall, Average Precision), across three well-known datasets (eBDtheque, DCM772, Manga109) and using visual inspections, we conclude that pre-trained self-supervised transformer models can competently outperform state-of-the-art approaches, which often require further fine-tuning to achieve comparable results. Moreover, the character identification module has been examined using neural networks and unsupervised learning. Following the qualitative study, it was determined that this task is not universally applicable across various comic books. Instead, it should concentrate on the characters within a single comic book or volumes within the same series.

SUBJECT AREA: Image processing, Document analysis systems, Document image processing, Physical and logical layout analysis

KEYWORDS: Comics, Object Detection, Object Segmentation, Panel Detection, Character Detection, Text Area Detection, Neural Networks, Transformers

ΠΕΡΙΛΗΨΗ

Τα κόμικς, που συνδυάζουν τέχνη με αφήγηση, συνεχίζουν να συναρπάζουν αναγνώστες, παραγωγούς κινηματογράφου και συλλέκτες, διατηρώντας τη γοητεία τους ως μια αγαπημένη μορφή οπτικής αφήγησης εδώ και δεκαετίες. Ο διαχωρισμός εικόνων στα κόμικς αποτελεί ένα κρίσιμο στοιχείο της ψηφιακής μεταμόρφωσης των κόμικς. Αξιοποιώντας ευριστικές μεθόδους, μοντέλα που βασίζονται σε νευρωνικά δίκτυα (YOLO) και καινοτόμες αρχιτεκτονικές transformer (GroundingDINO, SAM), η έρευνά μας στοχεύει στον αυτόνομο διαχωρισμό των σελίδων κόμικς στα βασικά τους συστατικά: καρέ, χαρακτήρες κόμικς και περιοχές κειμένου. Για το σκοπό αυτό, εκπαιδεύσαμε περαιτέρω τα μοντέλα YOLOv5 και YOLOv8 για να εντοπίσουν αυτά τα συστατικά, ενώ τα μοντέλα βασισμένα σε transformers χρησιμοποίησαν προτροπές για την ανάκτησή τους. Συγκρίνοντας την απόδοσή τους, με βάση καθιερωμένες μετρικές (Precision, Recall, Average Precision), σε τρία γνωστά σύνολα δεδομένων (eBDtheque, DCM772, Manga109) και χρησιμοποιώντας οπτικές επιθεωρήσεις, καταλήγουμε στο συμπέρασμα ότι τα προεκπαιδευμένα μοντέλα self-supervised transformers μπορούν να ξεπεράσουν επαρκώς τις σύγχρονες μεθόδους, που συχνά απαιτούν περαιτέρω προσαρμογή για να επιτύχουν συγκρίσιμα αποτελέσματα. Επιπλέον, το σύστημα αναγνώρισης χαρακτήρων έχει εξεταστεί χρησιμοποιώντας νευρωνικά δίκτυα και μη εποπτευόμενη μάθηση. Μετά από τη ποιοτική μελέτη, διαπιστώθηκε ότι αυτό το έργο δεν μπορεί να εφαρμοστεί καθολικά σε διάφορα κόμικς. Αντίθετα, θα πρέπει να επικεντρώνεται στους χαρακτήρες ενός μεμονωμένου κόμικ ή σε τόμους της ίδιας σειράς.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία Εικόνας, Συστήματα ανάλυσης εγγράφων, Επεξεργασία εικόνας εγγράφων, Ανάλυση φυσικής και λογικής διάταξης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Κόμικς, Ανίχνευση αντικειμένων, Τμηματοποίηση αντικειμένων, Ανίχνευση πλαισίων, Ανίχνευση χαρακτήρων, Ανίχνευση περιοχής κειμένου, Νευρωνικά δίκτυα, Transformers

Στους γονείς μου Ζωγραφία και Παναγιώτη

ACKNOWLEDGMENTS

Θα ήθελα καταρχήν να ευχαριστήσω τους καθηγητές και ερευνητές κ. Παπαβασσιλείου Βασίλη και κ. Κατσούρο Βασίλη για την επίβλεψη αυτής της μεταπτυχιακής εργασίας και για την ευκαιρία που μου έδωσαν να την εκπονήσω σε συνεργασία με το ερευνητικό κέντρο Αθηνά, καθώς και τον κ. Γάτο Βασίλη για την συμβολή του στην τριμελή επιτροπή και τις στοχευμένες επισημάνσεις του. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου για την εμψύχωση και ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια καθώς και τους φίλους μου, οι οποίοι ήταν κοντά μου σε κάθε δυσκολία.

CONTENTS

1 INTRODUCTION **23**

1.1 Objectives of Master Thesis . . . **23**

1.2 Problem Statement . . . **24**

1.3 Organization . . . **24**

2 RELATED WORK **25**

2.1 Datasets . . . **25**
2.1.1 eBDtheque . . . 25
2.1.2 Digital Comic Museum 772 (DCM772) . . . 25
2.1.3 Manga109 . . . 26
2.1.4 ICDAR2019-FGC . . . 26

2.2 Panel Detection . . . **26**

2.3 Character Detection / Identification . . . **27**

2.4 Text Area Detection . . . **27**

3 THEORETICAL BACKGROUND **29**

3.1 Image Processing-Filters . . . **29**
3.1.1 Average/ Mean filter . . . 30
3.1.2 Gaussian Blur Filter . . . 30
3.1.3 Median filter . . . 30

3.2 Morphological Operations . . . **30**
3.2.1 Erosion . . . 31
3.2.2 Dilation . . . 31
3.2.3 Opening . . . 31
3.2.4 Closing . . . 31

3.3 Edge Detection Algorithms . . . **32**
3.3.1 Sobel . . . 32
3.3.2 Canny . . . 32

3.4 Object Detection Models . . . **33**

3.4.1 You Only Look Once (YOLO) . . . 33
3.4.2 Segment Anything Model (SAM) . . . 35
3.4.3 GroundingDINO . . . 37

4 SYSTEM OVERVIEW 41

4.1 Neural Network-based approach . . . 42

4.2 Transformer-based approach . . . 42

4.3 Panel Detection . . . 42

4.4 Character Detection . . . 44

4.5 Text Area Detection . . . 45
4.5.1 Heuristic text detection method . . . 45

4.6 Character identification . . . 48

5 EXPERIMENTS 53

5.1 Panel Detection . . . 55

5.2 Character Detection . . . 55

5.3 Text Area Detection . . . 56

5.4 Character identification . . . 57

6 CONCLUSIONS AND FUTURE WORK 61

REFERENCES 65

LIST OF FIGURES

3.1 YOLOv5 high level architecture [1] 34
3.2 Example of how the YOLO model works. It divides the image into an S × S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. [2] 35
3.3 YOLO history flowchart [3]. 36
3.4 Segment Anything Model (SAM) overview [4]. 36
3.5 Details of the lightweight mask decoder [4]. 37
3.6 The process of creating the SAM training dataset [4]. 37
3.7 GroundingDINO architecture [5]. 38

4.1 System pipeline 41
4.2 Panel Detection on eBDtheque (1st row), DCM772 (2nd row) and Manga109 (3rd row). The first column contains the dataset labels, the second the results of YOLOv8 and the third the results of GroundingDINO. 43
4.3 Character Detection on eBDtheque (1st row), DCM772 (2nd row) and Manga109 (3rd row). The first column contains the dataset labels, the second the results of YOLOv8, the third the results of GroundingDINO and the fourth the results of GroundingDINO & SAM. 44
4.4 Text Area Detection on eBDtheque (1st row), DCM772 (2nd row), Manga109 (3rd row). The first column contains the dataset labels of the text lines/area (except for the DCM772, which does not have text labels), the second the results of our heuristic approach, the third the results of GroundingDINO, and the fourth the results of GroundingDINO & SAM. 46
4.5 Heuristic text detection pipeline 47
4.6 The input image on the left and the masked output on the right, having visible only the detected text areas using heuristic text detection. 48
4.7 The binary and inverse binary image using k-means clustering. 49
4.8 The component analysis of each binary image. 50
4.9 Refinement of text area. 51

5.1 Clusters with similar characters 57
5.2 Clusters with similar characters 58
5.3 Clusters with confused characters 59
5.4 Clusters with confused characters 60

LIST OF TABLES

5.1 Panel and Character detection Precision/Recall for YOLO, GroundingDINO and GroundingDINO & SAM for DCM772 (72 images on test set based on [6]) 54
5.2 Panel, Character and Text Area detection AP@50 YOLO and GroundingDINO with previous work based on [7] for Manga109 same test set (880 images) 54
5.3 Panel, Character & Text Area detection Precision/Recall GroundingDINO with previous work for eBDtheque. The results are for the whole dataset. . 54
5.4 Panel and Character detection Precision/Recall for eBDtheque test set for YOLO, GroundingDINO, GroundingDINO & SAM compared [8][6] used 5-fold cross-validation. 55
5.5 Text area detection on eBDtheque dataset pixel-level metrics. 56

1. INTRODUCTION

Comic books have been a popular form of visual storytelling, captivating audiences with their unique blend of art and narrative. Comics had an extremely high penetration among readers in the previous decades and are still popular to either readers, cinema producers, or collectors. Many studies [9] [10] [11] have shown that comics help young children develop critical thinking, underscoring their significance for subsequent developmental stages. With the advent of digital platforms and the increasing digitization of media, the study and analysis of comic images have gained newfound significance. Comic images present a compelling challenge in image processing and computer vision due to their intricate artistic styles, diverse layouts, and the fusion of textual and visual content. Comics have a unique layout format that presents additional difficulties in specifying a query and finding results at the page level.

Comic image segmentation, which divides a comic page into meaningful regions, is key to unlocking a deeper understanding of a comic narrative's visual and textual elements. Accurate segmentation enables extracting individual panels, comic characters, faces, speech balloons, captions, and links between characters and balloons, facilitating various applications. Some examples of systems on comic books are comic translation [12], indexing [13], adaptive display on various devices (smartphones, tablet, laptops, etc.) [8], improved resolution [14] and even the automatic comic book generation [15].

1.1 Objectives of Master Thesis

The specific objectives of this master's thesis are as follows:

1. Literature Review: A comprehensive survey of the existing literature will be conducted to understand the state-of-the-art techniques in comic image analysis and indexing. This review will serve as a foundation for the research and identify gaps in current approaches.

2. Methodology Development: Novel image processing methodologies will be proposed and developed to address the unique challenges posed by comic image segmentation. These methodologies will encompass various aspects, including object detection, region clustering, text extraction, and pattern recognition.

3. Dataset Creation: Finding and reconstructing a representative dataset of diverse comic images is an important task. The dataset should be compiled, encompassing different genres, artistic styles, languages, and layout compositions. This dataset will be crucial for training and evaluating the developed techniques.

4. Evaluation: The proposed methodologies will be rigorously evaluated using quantitative and qualitative metrics. Measures such as Intersection over Union, Precision, Recall, mean Average Precision, and visual coherence performance.

5. Future steps: The research in the context of this master thesis, could contribute to the development of a comprehensive framework that will ultimately facilitate the generation of comics in virtual reality or enhance the comic-reading experience by infusing emotion and diverse voices. This innovation is particularly aimed at ensuring that individuals who face challenges in reading can equally derive enjoyment from comics. To achieve this objective, particular attention will be given to delineating the crucial stages of development.

1.2 Problem Statement

This master thesis aims to investigate the boundaries of comic image segmentation using computer vision principles and the latest and promising transformer-based models. The central goal is to compare and test state-of-the-art (i.e., YOLO) and new well-presented models (i.e., GroundingDINO & SAM) for autonomously identifying and separating distinct elements within comic pages. Specifically, the proposed pipeline identifies the panels, the characters, and the locations of dialogues and narratives. In addition, a heuristic approach will be presented and tested for efficiency for the text area detection. Moreover, character re-identification will be examined using neural networks and Machine learning models. Last but not least, the research of this master thesis is accepted and will be presented at the coMics ANalysis, Processing and Understanding (MANPU) workshop of the ICDAR 2024 conference, in Athens, Greece.

1.3 Organization

The structure of this master's thesis is as follows: Chapter 2 begins with thoroughly examining the Related Work on Panel, Character, Text Detection, and the available datasets, where it contextualizes its research within the existing scholarly landscape, addressing how datasets have been previously utilized and the methodologies employed for detecting various comic elements. The next section focuses on the 'Theoretical background' of the methods used in the proposed system. Following this foundation, the 'System Overview' section describes the proposed system architecture, highlighting the integration of neural networks and transformer models to innovate comic decoding. The 'Experiments' section rigorously evaluates the system's performance through various tests and scenarios, offering quantitative and qualitative analyses to substantiate the research claims. Finally, the master thesis concludes with 'Conclusions and Future Work', summarizing the key findings and the implications of this study for the field of comic decoding and proposing directions for future research to build upon the groundbreaking work presented.

2. RELATED WORK

Comic images are a non-ordinary part of document images. They contain small images as scenes of a story, called panels; inside them are the comic characters that ”talk” to each other, think something, or even do an action. So, digitizing a comic book should address multiple tasks (panel detection, character detection and identification, text detection and recognition). Many previous studies focused on specific tasks to propose an enhanced solution. However, [13] presents an end-to-end comic indexing tool based on deep learning. Its workflow involves panel detection, character/face detection, balloon localization and association with characters, and text recognition. The produced annotations are stored in an XML file following the Comic Book Markup Language.

2.1 Datasets

To develop modules for addressing these tasks, it is essential to utilize labeled data for training and evaluation. The following sections provide descriptions of the most renowned datasets that are accessible. Each dataset exhibits a label format and aligns with a subset of comic-related tasks.

2.1.1 eBDtheque

eBDtheque [16] is the most compact dataset that contains labels for all comic tasks. The main drawback is that the images are insufficient for training deep-learning models. It consists of 100 images, each paired with a svg file. Annotations in this dataset cover four classes: Panel, Balloon (with tailDirection), Character, and Line (with textType and text inside). The dataset contains 100 images, 850 panels, 1550 characters, 1092 balloons, and 4691 text lines. In our work, we used Version 3 - July 2019[1].

2.1.2 Digital Comic Museum 772 (DCM772)

DCM772 [13] comprises 772 annotated images sourced from 27 comics available in the Digital Comic Museum. Each image is accompanied by a text file containing annotations in the format ”class id and bounding box coordination”. The dataset includes annotations for Panel, Character, and Face classes (4470 panels, 8385 characters, 5438 faces), which are publicly available[2]. The images could be downloaded from the Digital Comic Museum[3] site.

[1] http://ebdtheque.univ-lr.fr/download/v3/
[2] https://gitlab.univ-lr.fr/crigau02/dcm-dataset/-/tree/master
[3] https://digitalcomicmuseum.com/

2.1.3 Manga109

Manga109 [17][18] encompasses annotations for 109 different manga volumes. Each book has an XML file containing annotations for four classes: Frame (panel), Face, Body, and Text. Annotations for all classes are represented as bounding boxes with their coordinates. This dataset offers a comprehensive collection of manga book annotations, facilitating research and applications in computer vision and comics analysis. In our work, we used the 2021 released version[4], which contains 10130 annotated images, 103850 panels, 157234 characters, and 147887 text areas.

2.1.4 ICDAR2019-FGC

The ICDAR2019-FGC[5] dataset serves a specific and intriguing purpose: to identify and match similar characters. It is part of the challenge of the ICDAR 2019 Competition in Sydney, Australia. This unique dataset is created by cropping character images from the DCM772 dataset, where each character is meticulously labeled from 1 to 100. This dataset is invaluable for tasks related to character identification, similarity analysis, and pattern matching. It offers a rich collection of character variations and is well-suited for researchers and practitioners in the field of computer vision and image analysis who seek to develop algorithms for character similarity assessment, text matching, or related applications. The careful curation and labeling of characters make the ICDAR2019-FGC dataset a valuable resource for advancing the capabilities of character identification systems and fostering innovative research in the domain.

2.2 Panel Detection

The most essential task is panel detection, which is considered the root task for all the following modules. In previous works, panels were identified using traditional image processing techniques like connected components analysis and line detection. However, some comics do not have a frame around each panel and these algorithms fall in fault. Therefore, [13] proposed an object detection model with convolutional neural networks (You Only Look Once, YOLO version 2) and trained it to identify the panels. To fine-tune the model, two approaches were followed based on the pre-trained weights from Pascal-VOC[19]. The first is Anchor Boxes Learning using k-means clustering (k=5) to find five representatives of bounding box shapes, and the second one is Representation Learning, which is a type of transfer learning. In Representation Learning, the backneck (Darknet) is trained on a different classification task and gets the weights to YOLO. Their research showed that traditional methods [20] have a slight advantage, on average, compared to YOLO. However, the more complex options, i.e., panels without a frame around them, can be detected correctly only with deep learning. Another work is the Comic-MTL [8]

[4] http://www.manga109.org/
[5] https://fgc.univ-lr.fr/task/

[6], which describes a model for multiple tasks at the same time (panel, face, character, narrative boxes, balloon & balloon association to its speaker). Multitask learning used in Comic-MLT reduces the computation time to analyze comic book images.

2.3 Character Detection / Identification

The next task is character detection, aimed at identifying the complete figures of comics' protagonists. For this task, [13] also, proposed YOLOv2. It was further trained with the DCM772 train set by adopting representative learning and anchor box learning strategies using page or panel images as input. Both traditional image processing and machine learning methods face difficulties in selecting features and generating heuristic rules for generalized characters because characters in each comic will vary significantly across comics (e.g., persons, animals, objects, etc.). Traditional methods are SIFT descriptor [21], Frequent Subgraph Mining (FSM) techniques for comic image browsing using query-by-example (QBE) model [22] and sketch-based query model [17]. Furthermore, a recent study [23] explored the enhancement derived from integrating two datasets (eBDtheque and Manga109) and examined the varying outcomes based on the resolution of page or panel inputs. This research implemented data augmentation techniques to equalize the sample representation between eBDtheque and Manga109. The findings indicate that models trained with a combined dataset exhibited superior performance. Additionally, it was concluded that training with panel-level data only (i.e., not including the whole comic page in training sets) could be more efficient and increase processing time but without a corresponding improvement in accuracy.

It is important to identify where are presented the same characters in a comic book with different clothes and gestures in each panel. One work based on this cartoon Character re-identification [24] used a histogram of oriented gradients (HOG) and efficient Subwindow Search (ESS) with Color Names (CN) features. CAST (Character Labeling in Animation) [25] introduces a self-supervised method for labeling characters in animations. It leverages motion tracking to learn unique character representations across frames, enhancing our understanding of animated content. Moreover, Zhang et al. propose an unsupervised approach for comic character re-identification [26]. Their method extracts facial and body features from static manga panels and utilizes clustering algorithms to achieve promising character recognition results.

2.4 Text Area Detection

Another critical task is text area detection. Some research was focused on speech balloon segmentation, others on text line detection, and others on text body detection. One previous work for balloon segmentation [27] used traditional techniques starting with adaptive Threshold Selection (for binarization), followed by Balloon Candidate Selection (selecting white connected components, as candidate balloons, if they enclose black connected

components, i.e. letters), and finally with Balloon Candidate Analysis (selecting thresholds for removing “false alarms”). Another research obtains deep learning models for balloon segmentation. The model in [28] combines the VGG-16 CNN model in a U-Net architecture to predict a pixel-wise segmentation. Moreover, [13] combines the DeepLabv2 model with thresholding [27] to reduce false positive detected areas and increase true positives specifically for open balloons. To this end, it keeps pixels that are proposed as inside a balloon from both methods. The deep learning model was tested with the eBDtheque dataset. However, the training dataset for the deep learning model is private.

3. THEORETICAL BACKGROUND

Computer vision is a field of Artificial Intelligence (AI), that empowers computers to extract meaningful insights from digital images and videos. In essence, computer vision operates similarly to human vision, albeit with humans enjoying a significant head start. Human vision benefits from a lifetime of experience to discern objects, estimate distances, detect motion, and identify abnormalities within images. Computer vision trains machines to perform these functions, but it accomplishes this feat in a much shorter timeframe using cameras, data, and algorithms rather than relying on retinas, optic nerves, and visual cortex. This expedited training process allows computer vision systems tasked with inspecting products or monitoring production processes to analyze thousands of items or operations per minute, swiftly identifying imperceptible defects or anomalies, thereby surpassing human capabilities [8]. It started with classical machine learning algorithms and evolved into deep learning models for many tasks. Some computer vision tasks are image classification, object detection, object segmentation, object tracking, content-based image retrieval, and some combination of images and natural language and speech.

3.1 Image Processing-Filters

Classical image processing techniques serve as the cornerstone of computer vision, playing a pivotal role in its historical development and continued relevance in the modern era. These fundamental methods have been instrumental in deciphering and manipulating visual data, paving the way for groundbreaking advancements in computer vision. From the early days of edge detection and image enhancement to more sophisticated tasks like object recognition and feature extraction, classical image processing techniques have been indispensable tools for researchers and engineers alike. As we delve into the rich history and enduring significance of these techniques, it becomes evident that they form the bedrock upon which the complex and ever-evolving field of computer vision has been built, offering both a historical perspective and crucial insights into its contemporary applications.

Filters play a pivotal role in the real of computer vision, serving a critical purpose in enhancing image quality and mitigating noise. In this context, various algorithms, including both linear and nonlinear approaches, are deployed to filter images, unlocking a multitude of essential capabilities in image processing. These filters are instrumental in tasks such as noise reduction and deblurring. It's worth noting that nonlinear filters exhibit distinct behavior from their linear counterparts, deviating from the principles of scaling and shift invariance and often yielding results that defy intuitive expectations. In this exploration of filters in computer vision, we will delve into their significance and the diverse array of applications they facilitate. Its filter contains the info of the neighbors' pixels using a mathematical operation. How many neighbors should be used is defined based on kernel size.

3.1.1 Average/ Mean filter

The most straightforward filter to implement is referred to as the average filter. Its primary function is to perform average smoothing on an image, and its name aptly reflects this operation. Essentially, each pixel within the image, denoted as 'I,' is substituted with the mean value derived from its neighboring pixels. This process effectively blends noise into the overall image. example

3.1.2 Gaussian Blur Filter

For many applications, Gaussian blur is regarded as an ideal blurring method, provided that the kernel support is sufficiently large to encompass the fundamental aspects of the Gaussian distribution. When applying a Gaussian filter with a square support, it possesses the property of being separable. In the context of 2D filtering, this means it can be broken down into a sequence of 1D filtering operations, both for the rows and columns of an image. In cases where the filter radius is relatively small, typically less than a few dozen, the most efficient approach to compute the filtering result involves a direct 1D convolution. It's worth noting that when convolution is performed, the result has a length of N+M-1, where N represents the size of the signal, and M denotes the size of the filter kernel (which is equivalent to 2r+1). In other words, the output signal ends up being longer than the input signal as a consequence of this convolution process. example

3.1.3 Median filter

One effective method for achieving noise reduction is through the utilization of the median filter, a non-linear digital filtering technique that is frequently employed to eliminate noise. This noise reduction step is a common preprocessing procedure aimed at enhancing the outcomes of subsequent processing steps, such as edge detection in an image. The reason behind the widespread adoption of median filtering in digital image processing lies in its ability, under specific conditions, to preserve the edges within images while simultaneously eliminating unwanted noise. The median filter operates as a non-linear, local filter, generating its output value as the middle element of a sorted array comprised of pixel values from the filter window. This unique characteristic of selecting the median value endows the filter with robustness against outliers, making it particularly effective for reducing impulse noise. example

3.2 Morphological Operations

Morphological operations have an important role in the classical computer vision. The erosion, dilation, opening, and closing are the basic morphological operations. They are

developed for binary images and take as input the original image and one kernel which decides the nature of the operation[1].

3.2.1 Erosion

Erosion, a fundamental concept in image processing akin to soil erosion, involves the gradual removal of the boundaries surrounding a foreground object, typically represented in white. This operation entails the movement of a kernel across the image, similar to the process of 2D convolution. When the kernel traverses the original image, a pixel will retain its value of 1 only if all the pixels within the kernel's span are also equal to 1. Otherwise, it gets eroded and is set to zero. Consequently, what transpires is the removal of pixels located near the object's boundary, with the extent of this removal contingent upon the size of the kernel. This effectively reduces the thickness or size of the foreground object, leading to a reduction in the white region within the image. Erosion proves to be a valuable technique for tasks such as eliminating small instances of white noise, which can be observed in the color space, as well as detaching two connected objects from each other.

3.2.2 Dilation

Dilation is essentially the opposite of erosion. In dilation, a pixel element is assigned a value of '1' if at least one pixel within the kernel is '1'. This operation results in the expansion of the white region in the image or an increase in the size of the foreground object. Typically, in scenarios like noise removal, erosion is followed by dilation. Erosion is used to eliminate white noise, but it also causes the object to shrink. To counter this, dilation is applied. Since the noise has been removed, it won't reappear, but the object's area increases, restoring its original size. Dilation is also valuable for connecting fragmented parts of an object, effectively repairing broken structures within the image.

3.2.3 Opening

The opening of A by B is obtained by the erosion of A by B, followed by dilation of the resulting image by B. It helps to remove noise in the background.

3.2.4 Closing

The closing of A by B is the opposite of the opening. Dilation of A by B and then erosion of the resulting image by B. It is used for removing small noisy points inside the foreground area.

[1] https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html

3.3 Edge Detection Algorithms

Edge detection algorithms are fundamental tools in image processing and computer vision. They play a crucial role in identifying and highlighting the boundaries and transitions within images, allowing computers to discern object shapes and structures. These algorithms are essential in various applications, from object recognition and image segmentation to medical image analysis and autonomous navigation. By detecting edges, these algorithms provide valuable information for understanding and interpreting visual data, making them a cornerstone of modern image analysis and computer vision systems. In this exploration of edge detection algorithms, we will delve into their principles, techniques, and applications, shedding light on their significance in the field of computer vision.

3.3.1 Sobel

The Sobel edge detection algorithm [29] is a discrete differentiation operator that computes an approximation of the gradient of the image. The outcome of the process indicates how abruptly or smoothly an image changes at a specific point, offering insights into the likelihood of that region representing an edge and the probable orientation of the edge. In practical terms, calculating the magnitude (likelihood of an edge) is more reliable and easier to interpret than determining the direction. Mathematically, the gradient of the image intensity function at each point is a 2D vector, with components representing derivatives in the horizontal and vertical directions. The gradient vector at a given point points to the direction of the largest possible intensity increase, and its length corresponds to the rate of change in that direction. Consequently, in a region of constant image intensity, the Sobel operator yields a zero vector, while at an edge, it produces a vector pointing across the edge from darker to brighter values.

3.3.2 Canny

The Canny edge detection algorithm [30], often regarded as the optimal edge detector, aimed to improve existing edge detection methods. Canny John established three key criteria for edge detection: low error rate to avoid missing edges and false responses, well-localized edge points, and a single response for each edge. To meet these criteria, the Canny edge detector begins by smoothing the image to remove noise and then calculates the image gradient to highlight areas with high spatial changes. It tracks along these regions, suppressing non-maximum pixels. Next, a process called hysteresis is applied, involving two thresholds: pixels below the lower threshold are considered non-edges, those above the higher threshold are marked as edges, and those in between are set to zero unless a path exists to a pixel with a gradient above the higher threshold.

3.4 Object Detection Models

The object detection is a chapter on computer vision challenges. It is a technique for computers to recognize and localize objects in an image or video, like humans. Object detection algorithms typically leverage machine learning or deep learning to mimic human recognition and produce meaningful results. Object detection applications[2] are shown in our daily life like number plate recognition, face recognition and mask detection for COVID-19 restrictions, object tracking at group games like baseball or cricket, self-driving cars as Tesla cars and robotics.

The state-of-the-art object detection algorithms, in recent years, are: YOLO [2], MaskR-CNN [31], FastRCNN [32]. However, traditional computer vision algorithms like image filtering always help with preprocessing steps and for solving problems without labeled data. What is more, the rapid development of Large Language Models (LLMs) especially after the ChatGPT[33] release, changed the computer vision and object detection area. Based on Transformer architecture, pretrained visual transformers were developed and released as open-source software. The results of visual transformers are impressive.

For the master thesis, we are focused on YOLO-architecture and Transformer-based machine learning models. So, in the next subsections, it will be described briefly the models that we used.

3.4.1 You Only Look Once (YOLO)

YOLO[2][3] model is a state-of-the-art model that was developed in 2015 by Facebook AI Research team. After that, many modifications and improvements were made creating different versions of YOLO with the latest one being in 2023 and maybe invented others in the next years. The architecture of the YOLO is based on Convolutional Neural Networks which extract features from the input image and fully connected layers that predict the output probabilities and coordinates. One high-level figure of the base architecture of the YOLO v5 is figure 3.1. The backbone of version 5 is a convolutional neural network CSPDarknet53 and of version 8 is EfficientRep. The neck is used to multi-scale the features using Spatial Pyramid Pooling for version 5 and Feature Pyramid Network in version 8 combined with PANet to improve multi-scale feature aggregation. The head contains MLP detection layers predicting at three scales in version 5 and Enhanced detection layers with an anchor-free mechanism for better performance in version 8.

Moreover, the Non-Maximum Suppression (NMS) is an important post-processing algorithm that is used after the region proposal step to eliminate duplicate bounding boxes and select the most relevant ones. The idea behind NMS is straightforward. It works by comparing the confidence scores of the proposed bounding boxes and eliminating the ones that overlap significantly with a higher-scoring bounding box. NMS greatly decreases the occurrence of inaccurate detection in object detection outcomes. False positives happen

[2]https://neptune.ai/blog/object-detection-algorithms-and-libraries

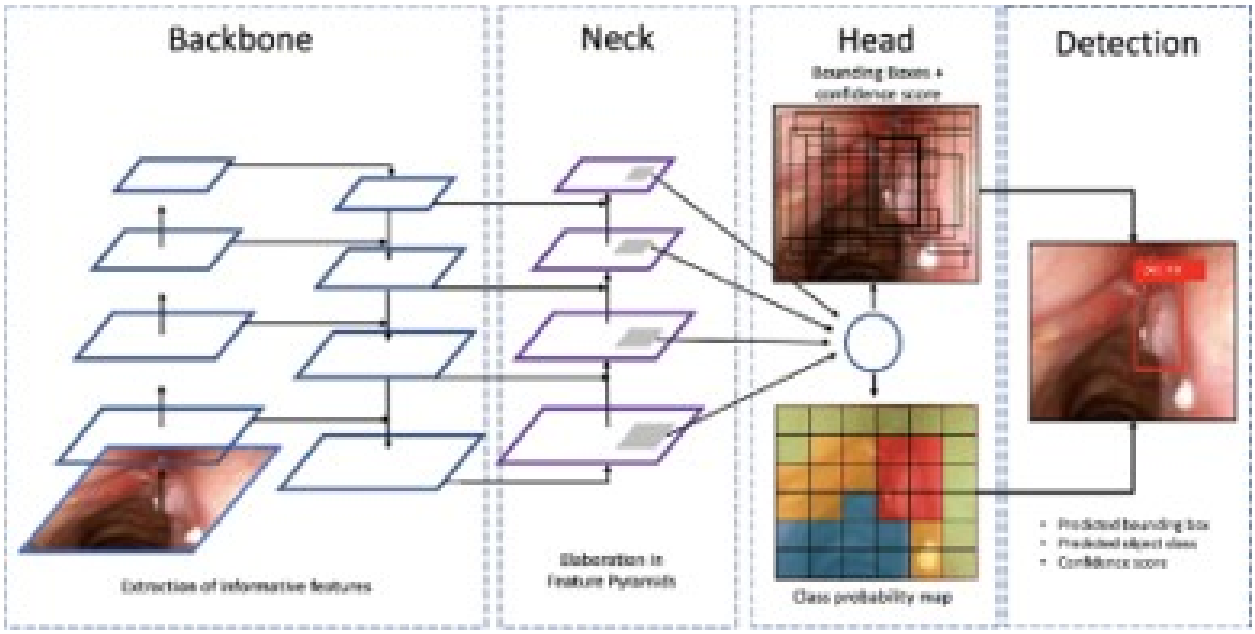

Figure 3.1: YOLOv5 high level architecture [1]

when a bounding box is created for an image region lacking an object. NMS tackles this issue by picking only the most pertinent bounding boxes related to the identified objects. Moreover, NMS aids in cutting down the computational workload of object detection algorithms by removing duplicate detections.

Additionally, in version 2 was introduced the anchor boxes element. Anchor boxes, also known as anchor priors or default boxes, are pre-defined bounding boxes with specific sizes, aspect ratios, and positions that are used as reference templates during object detection. These anchor boxes are placed at various positions across an image, often in a grid-like pattern, to capture objects of different scales and shapes. During training and inference, anchor boxes are used to predict the locations and shapes of objects relative to these reference boxes[3].

One timeline with the YOLO models was depicted in Fig. 3.3. The different versions were facing a trade-off between speed and accuracy. YOLO versions, including YOLOv2 (YOLO9000) and YOLOv3, refined the real-time capabilities by introducing anchor boxes, pass-through layers, and a multi-scale feature extraction architecture. As the YOLO framework evolved with models like YOLOv4 and YOLOv5, innovations such as new network backbones and optimized training strategies led to significant accuracy gains without compromising real-time performance. From YOLOv5 onwards, official models offer a fine-tuned trade-off between speed and accuracy, providing different scales tailored to specific applications and hardware. These versions often include lightweight models optimized for edge devices, prioritizing reduced computational complexity and faster processing times over absolute accuracy.

[3]https://medium.com/@nikitamalviya/object-detection-anchor-box-vs-bounding-box-bf1261f98f12

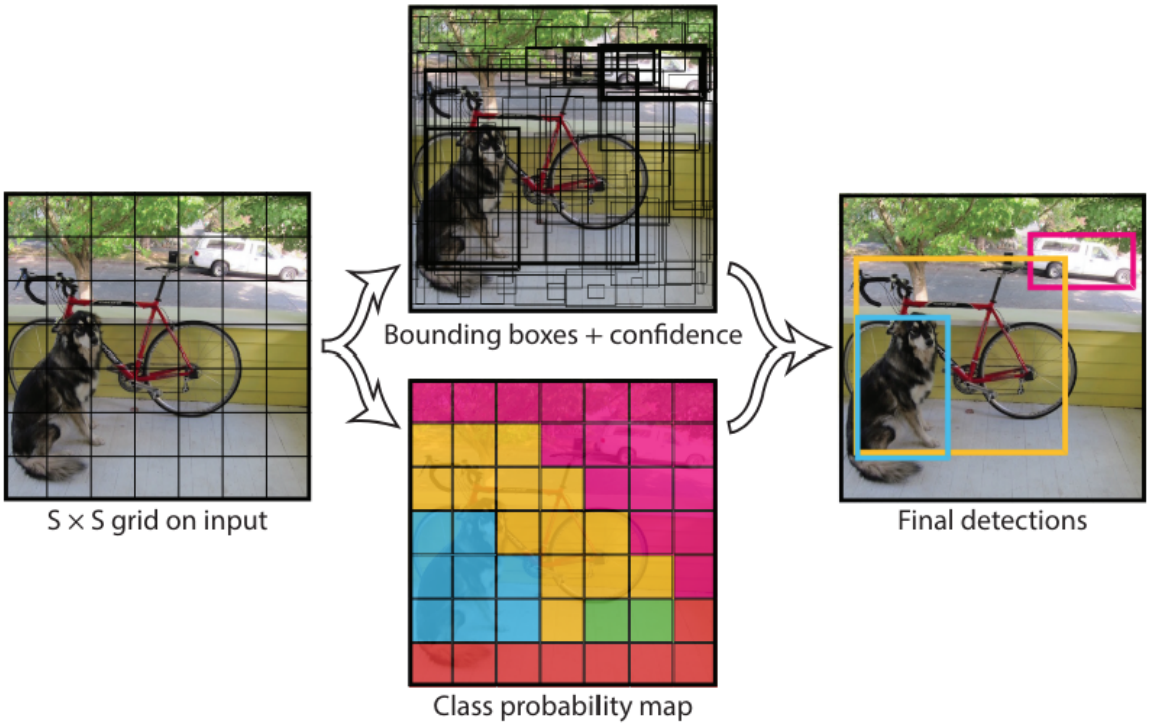

Figure 3.2: Example of how the YOLO model works. It divides the image into an S × S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. [2]

3.4.2 Segment Anything Model (SAM)

In April 2023, Meta AI introduced a robust model for object detection and segmentation known as SAM, short for the Segment Anything Model[4]. SAM has caused a significant revolution in the field of object segmentation due to its ability to address zero-shot challenges. Consequently, SAM eliminates the need for retraining for each task that is content-specific. As far as the SAM architecture is concerned, the main components are an image encoder, a prompt encoder, and a fast mask decoder. An overview is shown in Fig. 3.4.

- Image encoder has the Masked Autoencoder (MAE) Visual Transformer (ViT) architecture. This means that the encoder in this approach is based on the ViT model, but it is applied exclusively to visible, unmasked patches. Like a standard ViT, this encoder projects patches linearly, adds positional embeddings and processes them using Transformer blocks. However, our encoder focuses on a small subset (e.g., 25%) of the complete patch set and excludes masked patches, eliminating the need for mask tokens. This design enables the training of large encoders with significantly reduced computational and memory requirements, with the complete patch set being managed by a lightweight decoder.
- Prompt encoder has a different architecture based on the type of prompt. For prompt

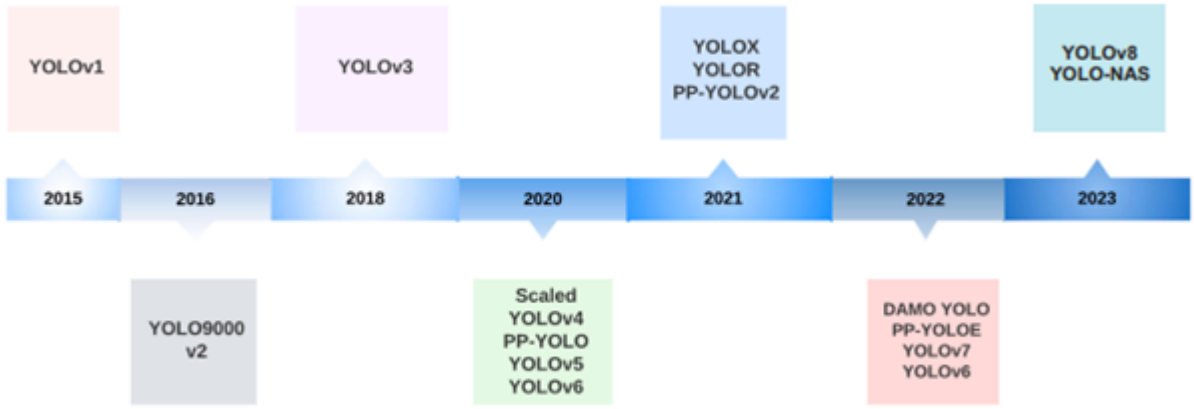

Figure 3.3: YOLO history flowchart [3].

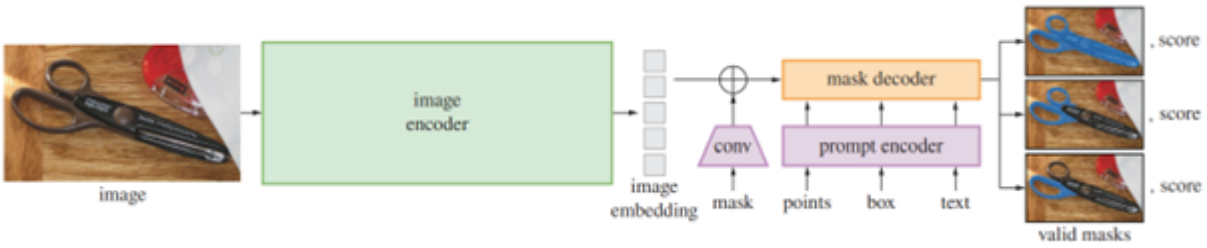

Figure 3.4: Segment Anything Model (SAM) overview [4].

points and boxes, it uses a model to represent them as positional encodings. For text prompts, a CLIP type of model is performed to combine text and image embeddings. Last but not least, for mask prompts, down-sampling is processed using convolution layers with GELU activation function and layer normalization. After that, element-wise addition runs with the image embeddings.

- Fast mask decoder (Fig. 3.5) consists of a Transformer decoder block with a dynamic mask detection head. It gets as input the prompt and image embeddings. The architecture of the decoder block consists of prompt self-attention and cross-attention in two dimensions (prompt-to-image and vice-versa) layer normalization and dropout of 0.1. Using cross-attention in both prompt-to-image and image-to-prompt directions helps the model to consider and weigh information from both the prompt and the image when updating embeddings. It's like making sure the decoder not only pays attention to the details of the prompt but also incorporates relevant features from the image. This two-way communication enhances the overall understanding of the context and improves the model's ability to generate more accurate and contextually relevant outputs. After two decoder blocks, the image embedding is upsampled with two transposed convolutional layers and passed through a 3-layer Multi-Layer Perceptron to dynamically calculate the output token with mask foreground probability at each image location.

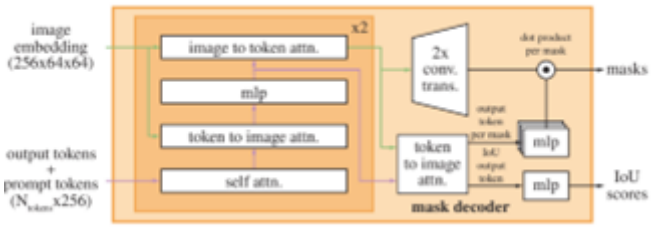

Figure 3.5: Details of the lightweight mask decoder [4].

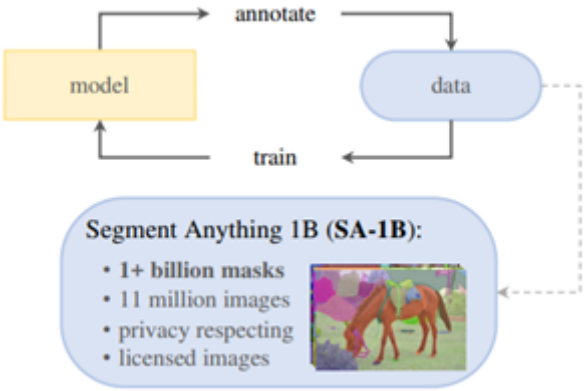

Figure 3.6: The process of creating the SAM training dataset [4].

Some other important aspects of SAM are based on the training process. First and foremost, the training dataset contains 11 Million images with more than 1 Billion masks. The masks are calculated using three stages: assisted-manual, semi-automatic, and fully automatic (Fig. 3.6). At the initial stage, SAM assists human annotators in the annotation process. In the second stage, SAM takes on a more active role by automatically generating masks for a subset of objects. This automation is triggered by providing SAM with probable object locations. The annotators then focus on annotating the remaining objects, contributing to a more diverse set of masks. The idea here is to leverage automation for efficiency while still maintaining human oversight and input. In the final stage, SAM operates fully autonomously. An interesting approach is taken here: SAM is prompted with a regular grid of foreground points across the image. This systematic approach results in the generation of approximately 100 high-quality masks per image, showcasing the ability of SAM to annotate images independently and systematically with a significant degree of automation.

Moreover, SAM learned and evaluated zero-shot problems in many different contents and tasks compared and outperformed the SOTA models.

3.4.3 GroundingDINO

GroundingDINO [5] was also developed by Meta AI in March 2023 aiming to detect bounding boxes from an image that are related to a text prompt. The Grounding DINO is an ex-

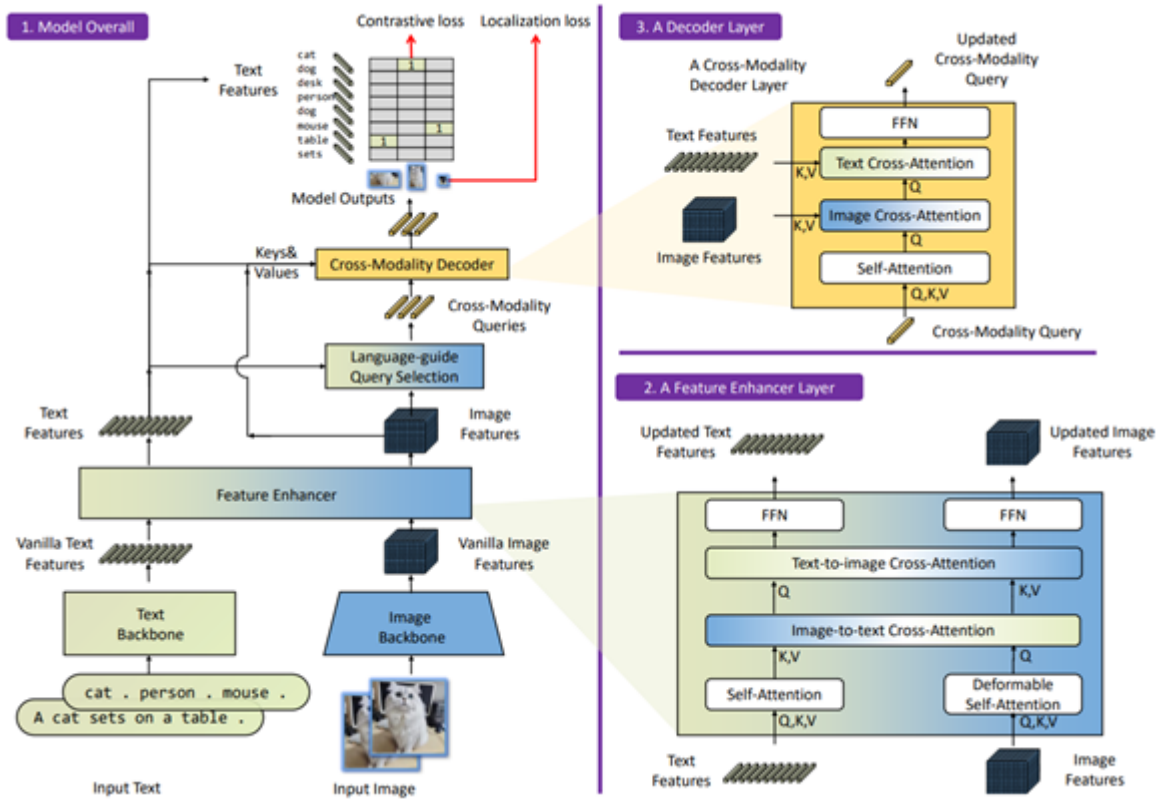

Figure 3.7: GroundingDINO architecture [5].

tended version of DINO [34]. The main difference between DINO and Grounding DINO is that Grounding DINO incorporates a grounding module that leverages textual information to improve object detection accuracy. Specifically, Grounding DINO uses a Sub-Sentence Level Text Feature to encode the input text in a way that eliminates the influence between different category names while keeping per-word features for fine-grained understanding. This helps the model better understand the input text and improve the accuracy of object detection. Additionally, Grounding DINO outperforms DINO on the zero-shot transfer setting and sets a new record on the COCO object detection benchmark without seeing any COCO images during training.

The architecture of Grounding DINO is depicted in Fig. 3.7. The main components of the model are:

- **Text Backbone** (BERT-like) to encode text input and then create Sub-Sentence Level Text Features that eliminate the influence between different category names while keeping per-word features for fine-grained understanding.
- **Image Backbone** (Swin-like) to encode image input.
- **Feature Enhancer** to fuse the vanilla image and text features obtained from the image and text backbones, respectively, for cross-modality feature fusion.

- **Language-Guided Query Selection** to select features that are more relevant to the input text as decoder queries to guide object detection.
- **Cross-Modality Decoder** to probe desired features from the two modal features using cross-modality text and image cross-attention layers and update the queries for object detection.

4. SYSTEM OVERVIEW

This chapter outlines the proposed methodology for the analysis of comic images. To begin with, the adopted pipeline was based on the workflow in [13]. Our approach is organized to encompass five fundamental components: panel detection, character detection, character identification, text area detection, and text recognition. These components collectively constitute the underlying framework for our strategy to analyze comic images, ensuring a comprehensive and efficient process. Notably, these tasks are interdependent, with panel detection of paramount importance. Each panel represents a distinct scene, typically containing one or more characters and speech balloons with text. Figure 4.1 illustrates the proposed pipeline's workflow. Initially, both panels and characters are identified. Subsequently, text areas of each panel are detected. Following this, text recognition via OCR is performed. Simultaneously, character detection results for each comic book are used to conduct clustering to identify the same characters across different panels.

In this master thesis, the tasks examined are the panel, character, and text area detection using different models and compared with previous works. Research was also for character identification. Text recognition can be the future work to finalize the comic analysis pipeline and get coherent content for each comic book. Having identified the critical components of a comic through this pipeline, additional components could be employed to bring comics into the digital era and improve the user-reader's experience. For instance, a machine translation subsystem could translate the identified and recognized text into another language and replace it within the designated area it should cover (i.e., into a "balloon or text caption"). Additionally, a text-to-speech component, separating the comics' text information into dialogues and narratives, and character identification could transform comics from an illustrated text into an audiovisual material/product and thus improve the user-reader's experience.

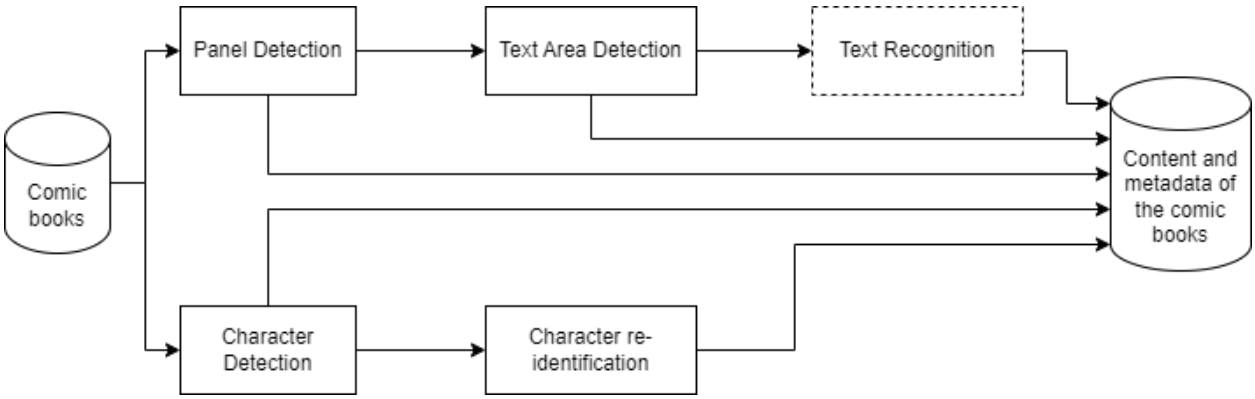

Figure 4.1: System pipeline

4.1 Neural Network-based approach

For the first approach, given the fact that the paper [13] used the YOLOv2 architecture to solve panel/character and face detection, we experimented with the fifth and eighth versions of YOLO [35]. The training dataset contains a combination of the three open datasets mentioned in section 2.1, which have the bounding boxes of both panels and characters to be used as ground truth labels to handle multiple classes and not each one detected with a different model. In particular, we used the same test set of DCM772 and Manga109 as shown in [6] (72 images), [7](880 images) and 10 of 100 images of eBDtheque randomly selected. The remaining images were used for the training phase (DCM772: 700 images, eBDtheque: 90 images, Manga109: 9250 images).

4.2 Transformer-based approach

The second approach focuses on two recently released state-of-the-art zero-shot Transformer-based models for object detection. Our research examined the pre-trained models Segment Anything Model (SAM) [4] and GroundingDINO[5], which have open-source code and pre-trained weights by the Meta AI team. SAM can separate an image into masks and use prompts to focus on a specific area. In our case, this model cannot be used stand-alone because the pre-trained weights of the model with the text prompt as input have yet to be publicly available.

GroundingDINO is closer to our problem. It gets an input, text prompt (e.g. ‘panel‘, ‘text‘, ‘character‘), and image and gives the bounding boxes of elements like the text prompt. Finding segmented masks of each detected component is valid in the character and text detection case. For that reason, we performed the SAM, giving as input the image and the GroundingDINO bounding boxes as prompt. We want to mention that the results of segmented masks highly depend on GroundingDINO results. If GroundingDINO does not find the bounding box, the segmented mask cannot be detected. However, it was noticed that sometimes, even though a box was found from the GroundingDINO, SAM could not find any mask. For that reason, we investigated the results of the GroundingDINO & SAM as a different model.

4.3 Panel Detection

Panel detection can be examined as an object detection process. Two types of model architectures were used to develop the detection model, described in the above sections 4.1, 4.2. First of all, a visual inspection was performed in the three datasets (Figure 4.2).

The results show that both the trained model YOLO and the pre-trained model GroundingDINO can detect panels highly accurately. The difference is that the trained YOLO model can be more generic in panels without a surrounding frame. However, the ground

Figure 4.2: Panel Detection on eBDtheque (1st row), DCM772 (2nd row) and Manga109 (3rd row). The first column contains the dataset labels, the second the results of YOLOv8 and the third the results of GroundingDINO.

truth is questionable for those cases, and we cannot criticize any of them as wrong. The GroundingDINO has been observed in the scanned images of the DCM772 (Fig.4.2 second row), which considers the whole page as a panel. This prompted us to investigate its results, removing the panels that fully involved the other panels. Furthermore, both models perform well in Manga, which has more unstructured panels.

4.4 Character Detection

The model architecture and processing we used are described in Section 4.1, 4.2. The trained YOLO model returns as output the bounding boxes of the detected panels and characters at the same time. Furthermore, the GroundingDINO runs without further fine-tuning, giving as input the text prompt = 'character' and returning the bounding box of the detected characters. Last, the GroundingDINO & SAM returns the segmented mask of the characters. Examples for each dataset are depicted in Figure 4.3.

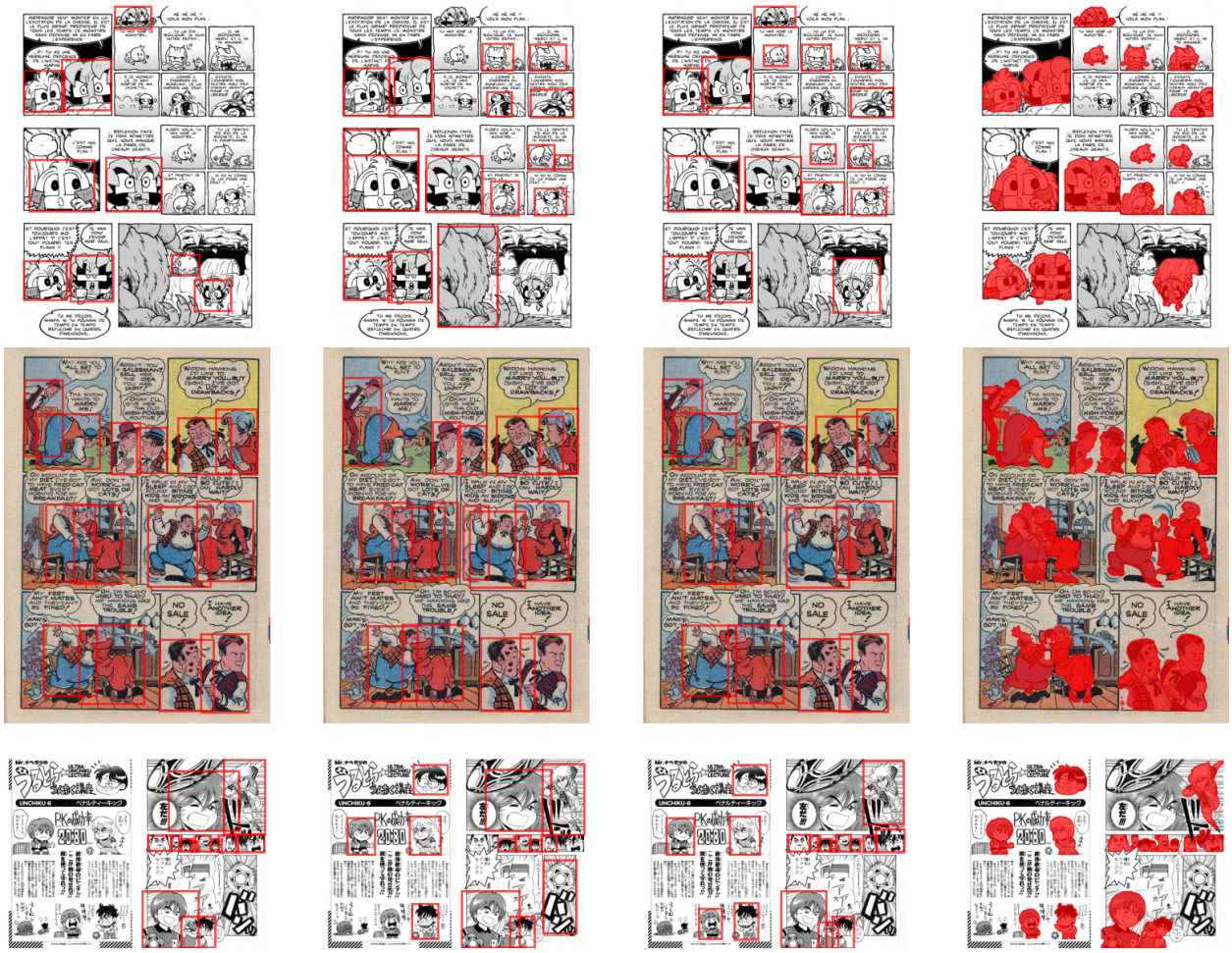

Figure 4.3: Character Detection on eBDtheque (1st row), DCM772 (2nd row) and Manga109 (3rd row). The first column contains the dataset labels, the second the results of YOLOv8, the third the results of GroundingDINO and the fourth the results of GroundingDINO & SAM.

The main characters of each image are detected by every model. Only some small characters in Fig. 4.3 in the first row were not detected and one character in the third row and third and fourth column. What is more, the models found additional characters that are not labeled in the original dataset (Fig.4.3 first and third row). It is worth mentioning that there are some missing objects in the ground truth datasets.

4.5 Text Area Detection

The primary challenge encountered pertains to the detection of text areas within images. In our endeavors, a significant limitation was the need for labeled images, attributed to the inadequacy of one of the three datasets (DCM772), which lacks annotations for text or speech balloon areas. Conversely, the eBDtheque dataset includes annotations for text lines and speech balloons, whereas the Manga109 dataset provides labels for text area regions differing from text lines or balloons. The challenge of training exclusively with the eBDtheque dataset is compounded by its limited volume of images, and training with Manga109 is problematic due to its specificity to the Japanese language, rendering generalization across different languages challenging. Consequently, we explored the use of GroundingDINO and GroundingDINO & SAM and introduced a heuristic text detection method inspired by the methodology presented in [27], which will be described deeply in the next section.

Some examples are shown in Figure 4.4. We can observe that transformers detect the text of all languages (i.e., French, Japanese, and English). The errors of the transformers concern a small text in Fig.4.4 first row, and a small false positive area in Fig.4.4 second row. As far as the heuristic approach is concerned, the Japanese language struggled to be identified. Moreover, the words above the first panel in Fig.4.4 first row cannot be predicted because the heuristic pipeline depends on panel detection and can identify only the text inside the panel. Furthermore, the narrative box in Figure Fig.4.4 third row on the panel above left does not match the specifications we perform in the heuristic approach (i.e., the letters have high contrast with the background), so the algorithm fails. Finally, some false positive masks are also presented.

4.5.1 Heuristic text detection method

The method is applied on each detected panel separately as depicted in Fig. 4.5. First, the panel is binarized to create a distinction between potential text areas (foreground) and the rest of the content (background). Connected components analysis is employed by using kernel density estimation to map out the distribution of component areas. This analysis facilitates the identification of patterns indicative of text (lobes in the pdf graph), with the search for local maxima and minima within the distribution aiding in pinpointing probable text regions. The algorithm applies to a multi-scale image pyramid, adjusting the scale of the image and repeating the binarization and analysis until greater than two text regions are detected. Once a sufficient quantity of potential text areas is detected, a specialized filter is applied for refinement. This filter aims to enhance the precision of the identified text regions. It operates by assessing the distinct connected components. These components are evaluated within a kernel. The size of this kernel is not fixed; it dynamically adjusts. The adjustment is based on the median area of the connected components.

In order to understand the steps of heuristic text detection, we can visualize the main tasks. Firstly, we can see an example shown in the Fig. 4.6. More deeply, for the panel bottom

Figure 4.4: Text Area Detection on eBDtheque (1st row), DCM772 (2nd row), Manga109 (3rd row). The first column contains the dataset labels of the text lines/area (except for the DCM772, which does not have text labels), the second the results of our heuristic approach, the third the results of GroundingDINO, and the fourth the results of GroundingDINO & SAM.

left, we binarized the panel using k-means of 2 clusters and got the result of Fig. 4.7. So, we have 2 images, one is the foreground (with text) and the other is the background (with the balloon). To identify which cluster is in the foreground we perform a connected component analysis based on the area of the components and the kernel density distribution of the components' area shown in Fig. 4.8. The first image is the binary image that performed the component analysis. The second plot is the histogram of the area of the connected components of the first image in the range of 0-400 pixel area. The third plot is the pdf graph and the last image is the mask of the connected components selected after applying the conditions. The conditions are to choose the components that have an area inside the second lobe of the pdf (between the local maximum and local minimum. After selecting the binary image that has more connected components that fulfill the conditions,

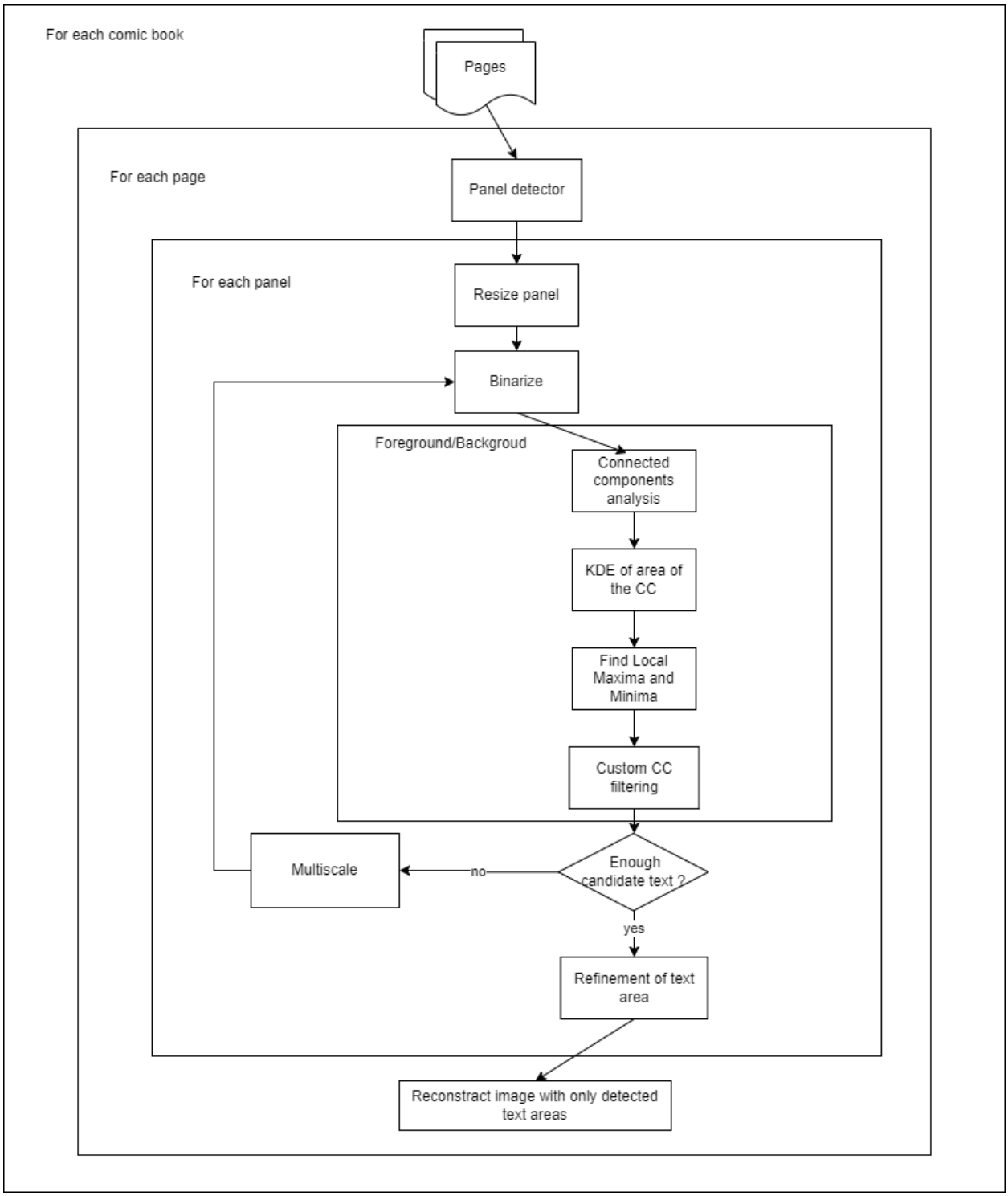

Figure 4.5: Heuristic text detection pipeline

we performed a final refinement of the results to remove false positive findings. Each connected component selected corresponds to a word character, so, if it is correctly detected there will be > 3 connected components one close to the other. Hence, we performed a

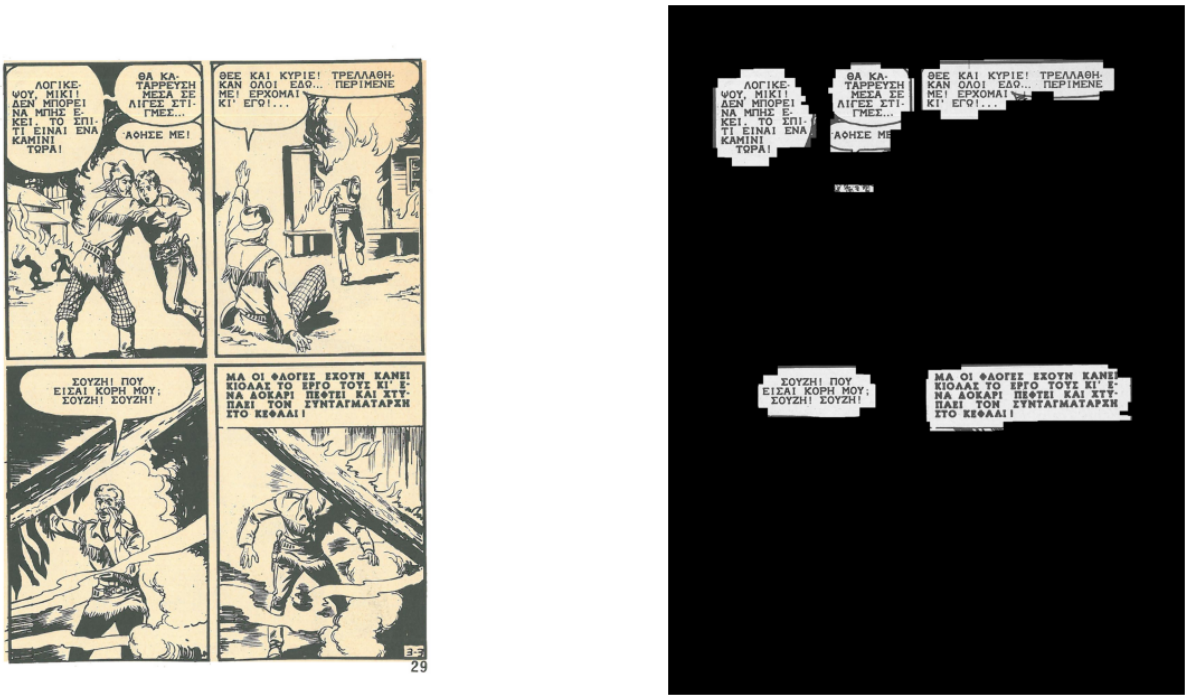

Figure 4.6: The input image on the left and the masked output on the right, having visible only the detected text areas using heuristic text detection.

count distinct filtering as shown in Fig. 4.9.

4.6 Character identification

The Character re-identification is a different task than the previous ones. Finding the same characters in different panels and pages, which may have different poses and clothes, is important. It is based on the recognition of the same detected characters. It will be treated as an unsupervised task. For that, we experimented with CNN-based feature extraction and clustering algorithms. The CNN model we used is the VGG16 pre-trained to the ImageNet. The features we used are the output of the VGG16 on the layer before the classification layer (4096 features) and then doing a dimensionality reduction using PCA the features were the 200 most important features. So, using the 200 features of each image, we performed K-means clusterings until we found out the most appropriate K number according to the elbow method. The dataset ICDAR2019-FGC (Section 2.1.4), we used, was designed for this task, to identify and match similar characters. Moreover, one other option to identify similar characters is to use as a reference a cropped character image and then try to find image areas that present the same extracted features in the same image or other images. In this way, it is easy to identify the areas that depicted the protagonists of the comic book.

Figure 4.7: The binary and inverse binary image using k-means clustering.

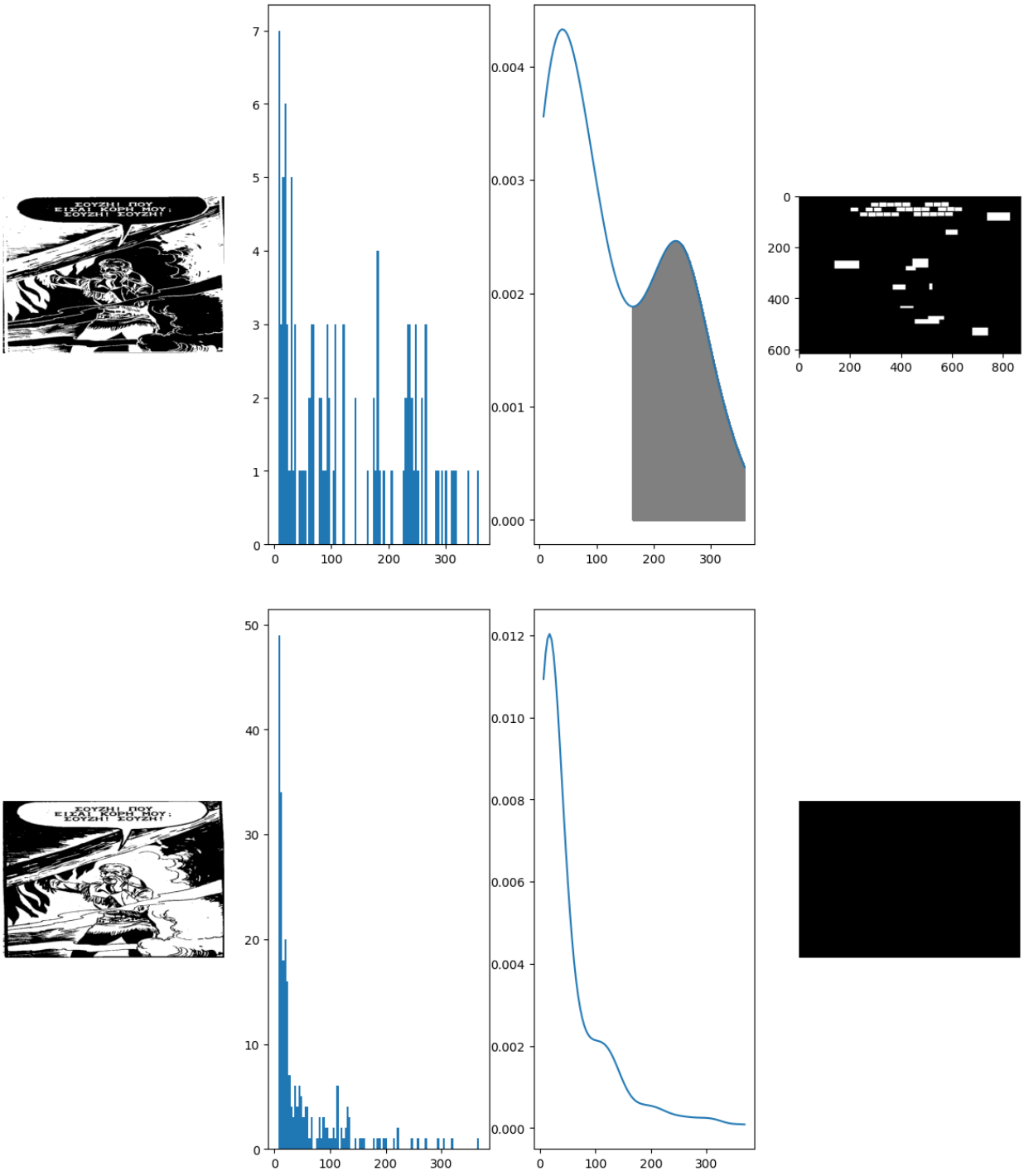

Figure 4.8: The component analysis of each binary image.

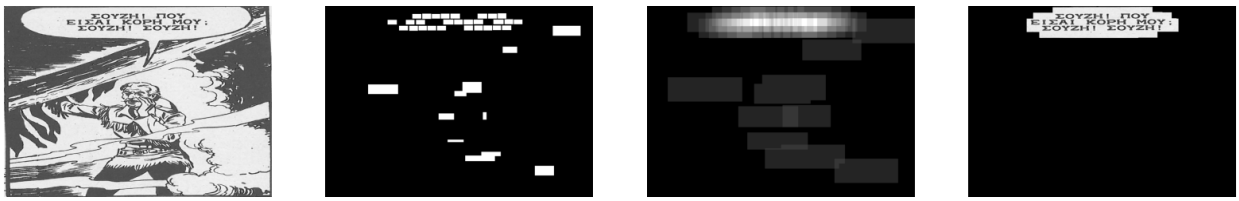

Figure 4.9: Refinement of text area.

5. EXPERIMENTS

The main concept of the experiments for the object detection tasks is to evaluate the convolutional neural network YOLO retrained in a comic-specific combined dataset, the zero-shot pre-trained transformer-based models, and a heuristic image processing approach in text area detection, compared with previous work.

For the neural networks, we trained a YOLOv5-large and YOLOv8-large model using a combined dataset of the three available datasets (eBDtheque, DCM772, Manga109) separating the same test set as the [13] and [7] mentioned for the DCM772 (72 images) and Manga109 (880 images), respectively. For the eBDtheque, 10 of 100 images were randomly selected for the test set. The training set was randomly separated into train and validation sets with a ratio of 90%/10% in each dataset. After transforming the ground truth labels to the YOLO input format, the model trained for 250 epochs, batch size 32, and resized the images with a maximum size of 256 pixels. The same model was introduced to identify panels and characters with different class identifiers.

The next model we tested on panel, character, and text area detection is the GroundingDINO. This model was used without further training on comic-oriented images. As input, the model gets the text prompt 'panel', 'character' and 'text' respectively and returns the bounding boxes that were similar to that prompt. We experimented using SwinB and SwinT weights, box, and text thresholds. After experiments, the thresholds selected are box threshold 0.35 and text threshold 0.25, the backbone of SwinB[1] and keep the results with confidence greater than 35%. Model was trained using data from O365[36], VG[37], RefCOCO[38], COCO[39], OpenImage[40], Cap4M[41] and ODinW-35[42].

The combined GroundingDINO and SAM model was also used to find the segmented characters and text areas. The pipeline of this model is to get the bounding boxes of the GroundingDINO predictions and passed to SAM as a prompt to find the proper mask. The parameters selected for SAM are the Visual Transformer(ViT) huge weights [2] and multimask output argument equal False to return only the best result for each bounding box.

To evaluate the performance of the models, the Precision and Recall metrics were calculated to compare them with previous work. We also used the Average Precision metric as PASCAL-VOC[19] mentioned. To provide comparable results, we adopted Intersection over Union (IoU) for calculating these metrics and set the success threshold to 50%. We performed a small change in this threshold for GroundingDINO & SAM, and the heuristic approach, because the masks are included in the bounding boxes, without covering all their areas. Experimentally, we decided to reduce the threshold to 30%.

[1] https://github.com/IDEA-Research/GroundingDINO/releases/download/v0.1.0-alpha2/groundingdino_swinb_cogcoor.pth

[2] https://dl.fbaipublicfiles.com/segment_anything/sam_vit_h_4b8939.pth

Table 5.1: Panel and Character detection Precision/Recall for YOLO, GroundingDINO and GroundingDINO & SAM for DCM772 (72 images on test set based on [6])

	Panel		Character	
Model	Precision	Recall	Precision	Recall
Rigaud [43]	86.78	74.84	-	-
Rigaud [44]	85.22	74.41	-	-
Nguyen-YOLOv2 [13]	84.75	86.62	-	-
Faster R-CNN [6]	92.10	93.21	78.93	65.25
Comic MTL - optimized [6]	96.84	97.76	76.21	67.56
YOLOv5	**98.25**	98.9	61.56	68.09
YOLOv8	97.41	**99.12**	**82.99**	63.58
GroundingDINO	90.97	92.97	79.77	84.04
GroundingDINO-post	93.17	92.97	-	-
GroundingDINO & SAM	-	-	77.8	**87.09**

Table 5.2: Panel, Character and Text Area detection AP@50 YOLO and GroundingDINO with previous work based on [7] for Manga109 same test set (880 images)

Model	Panel	Character	Text Area
Faster R-CNN	96.1	63.9	23.8
SSD300	**97.1**	79.1	82.0
YOLOv2	90.2	46.9	64.6
SSD300-fork	96.9	**79.6**	**84.1**
YOLOv5	80.5	58.5	-
YOLOv8	83.5	61.9	-
GroundingDINO	85.2	77.6	45.9
GroundingDINO & SAM	-	78.7	59.8

Table 5.3: Panel, Character & Text Area detection Precision/Recall GroundingDINO with previous work for eBDtheque. The results are for the whole dataset.

	Panel		Character		Text Area	
Model	Precision	Recall	Precision	Recall	Precision	Recall
Nguyen-YOLOv2 [13]	83.44	58.96	-	-	-	-
Rigaud [43]	86.55	81.24	40.5	21.6	-	-
SSD300-fork [7]	73.30	76.40	58.0	42.2	-	-
Heuristic approach	-	-	-	-	33.64	61.07
GroundingDINO	92.23	**83.76**	83.52	67.59	**93.92**	**82.77**
GroundingDINO-post	**93.52**	83.18	-	-	-	-
GroundingDINO & SAM	-	-	**86.26**	**72.47**	74.61	80.47

Table 5.4: Panel and Character detection Precision/Recall for eBDtheque test set for YOLO, GroundingDINO, GroundingDINO & SAM compared [8][6] used 5-fold cross-validation.

	Panel		Character	
Model	Precision	Recall	Precision	Recall
Comic MTL [8]	73.19	76.95	71.79	62.17
Faster R-CNN [6]	91.52	90.77	71.23	61.56
Comic MTL-optimized [6]	92.11	90.91	71.79	62.17
YOLOv5	95.08	89.23	63.91	**92.39**
YOLOv8	**98.39**	**93.85**	**82.18**	90.22

5.1 Panel Detection

The models we evaluate for panel detection in the three datasets are YOLOv5, YOLOv8, GroundingDINO and GroundingDINO combined with a post-processing step to reduce this error by removing the proposed panels that fully involved the others. The results are presented in Tables 5.1, 5.3, 5.4, 5.2 comparing with previous work.

In Tables 5.1 and 5.2 the results concern evaluation on the same test set as the previous work mentioned. In DCM772 (Table 5.1), YOLO models have the highest Precision and Recall, and GroundingDINO has remarkable results, too. Only Comic MTL outperformed GroundingDINO, but it should noticed that Comic MTL - optimized, Faster R-CNN, and YOLOv2 are trained using the DCM772 training dataset. Regarding the results on Manga109 (Table 5.2), it is obvious that the dedicated models (e.g. SSD300) outperform the others. Nevertheless, GroundingDINO has slightly better results than YOLO models, aside from the fact that it has never seen Manga images.

Table 5.3 shows the metrics about the 100 images of eBDtheque, and the comparison is with models that are not trained using this dataset. We can clearly identify that GroundingDINO has better results, and the exclusion of the identified panels that involved others contributes on reducing false positives (precision increased by 1.2%) while removing only a few true positive predictions (recall reduced by 0.6%).

Table 5.4 shows the results of our models in the test set compared to the 5-fold results of the other models. Obviously, YOLO attained higher scores but it should be noted that their evaluation is based on only 10 comic images. However, because the other models performed cross-validation and our results do not, our results are not highly reliable. GroundingDINO & SAM are missing, since it does not make sense to evaluate them on these 10 images. We have evaluated them on the whole eBDtheque dataset.

5.2 Character Detection

For the character detection, we evaluate YOLOv5, YOLOv8, GroundingDINO, and GroundingDINO & SAM in the three datasets. Table 5.3, 5.4, 5.1, 5.2 present the results. The

GroundingDINO & SAM outperforms both in precision and recall the other models in the eBDtheque dataset, has a higher F1 score in DCM772, and is closely (-0.9%) to the best-performing model (SSD300-fork). Once more, we mention that SSD300-fork is trained specifically for Manga images, while GroundingDINO has never seen Magna comics.

As we mentioned in the panel detection, Table 5.4 is not highly reliable for our results because our test set is small and the previous work performed cross-validation. Also, the SSD300-fork is trained specifically for Manga images, so the results have a benefit. The SAM model helps GroundingDINO to accelerate its performance. Our YOLO models also have a good position after GroundingDINO, except for the Manga109.

5.3 Text Area Detection

The last component investigated is text area detection. We evaluate the results of the heuristic approach compared with GroundingDINO and GroundingDINO & SAM on the eBDtheque dataset. The eBDtheque has labels about the text lines and the speech balloons. On the other hand, our models locate the text region either as a bounding box (GroundingDINO), or a segmented mask of (heuristic approach and GroundingDINO & SAM). To compare the results with the ground truth, we applied morphological closing on the textlines in order to merge them in case they are relatively close (i.e. less than 10 pixels). The first approach compares the labels and results using the IoU threshold of more or equal to 50% as in the previous tasks (See Table 5.3). The second one uses pixel-wise metric segmentation accuracy, i.e., the ratio of true positives to the sum of true positives, false positives, and false negatives (See Table 5.5). For that comparison, we observed that GroundingDINO and GroundingDINO & SAM managed to detect a good amount of text area. In contrast, the heuristic approach underperformed, finding out around many false positive components (low precision in Table 5.3) and many false positive and false negative pixels in Table 5.5. So, it may detect text but not the accurate text area mask close to the labels.

Furthermore, the GroundingDINO and GroundingDINO & SAM evaluated in Manga109 using AP getting around 60% compared with SSD300-fork that achieved 84.1%.

Table 5.5: Text area detection on eBDtheque dataset pixel-level metrics.

Model	Segmentation accuracy	Precision	Recall
Heuristic	31.78	47.66	48.82
GroundingDINO	**66.49**	**93.70**	69.60
GroundingDINO & SAM	60.30	70.84	**80.21**

5.4 Character identification

For this task, we examined extracting characters using a character detector from the previous step and using a pre-trained convolutional neural network (VGG16) we create encoded features that represent the images, do dimensionality reduction using Principal Component Analysis (PCA) method (from 4096 features to 200) and perform a K-means clustering exploring the best K using elbow criteria to separate them into groups.

Some clusters are coherent results as shown in the Fig. 5.1, 5.2 and some others are not the same character 5.3, 5.4. Mainly in most clusters, we do not have 100% the same characters but it seems that they have similarities in colors or contain a character which is close in many images with the main character of the cluster (e.x. in couples we have a cluster with the main character the woman but have some other images with the man near with her).

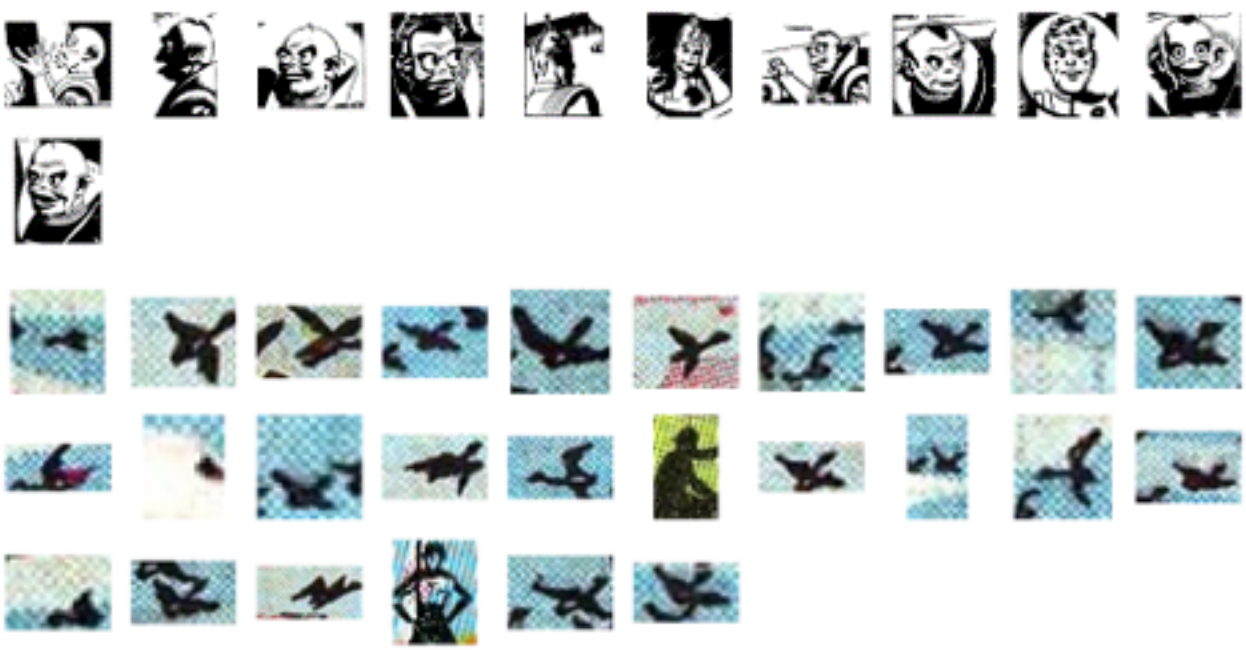

Figure 5.1: Clusters with similar characters

From my point of view, this task cannot be applied across different comic books but should focus on the characters of a comic book, or books of the same series. In addition, to the best of my knowledge, there is no related labeled dataset. However, I applied the proposed approach to the available dataset, which contains characters from different comic books together, in order to get an idea of what could be achieved by combining extracted features and clustering algorithms. The results are not really satisfying but if we have only the character of one book the separation would be easier and the model may have better performance.

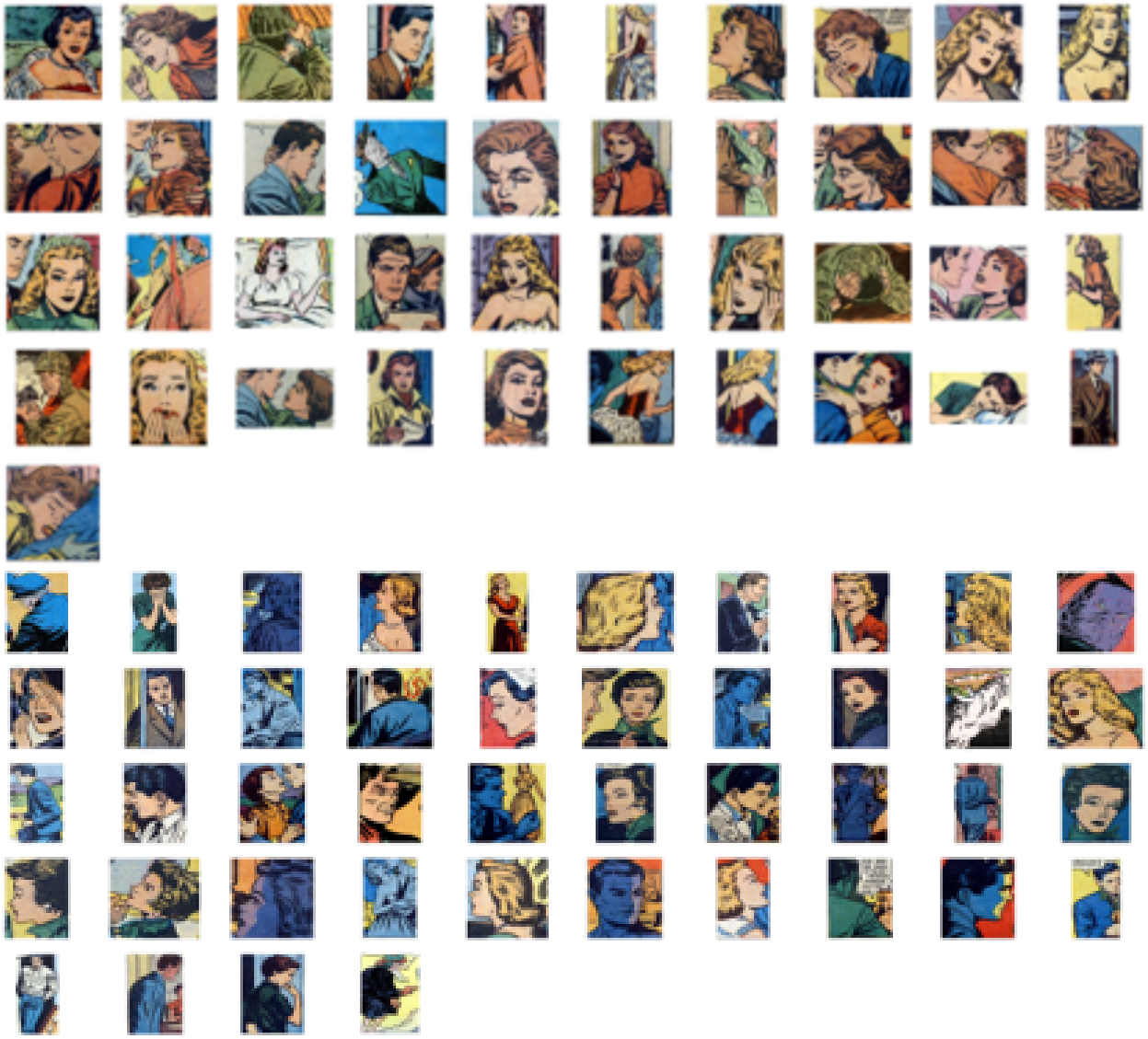

Figure 5.2: Clusters with similar characters

Figure 5.3: Clusters with confused characters

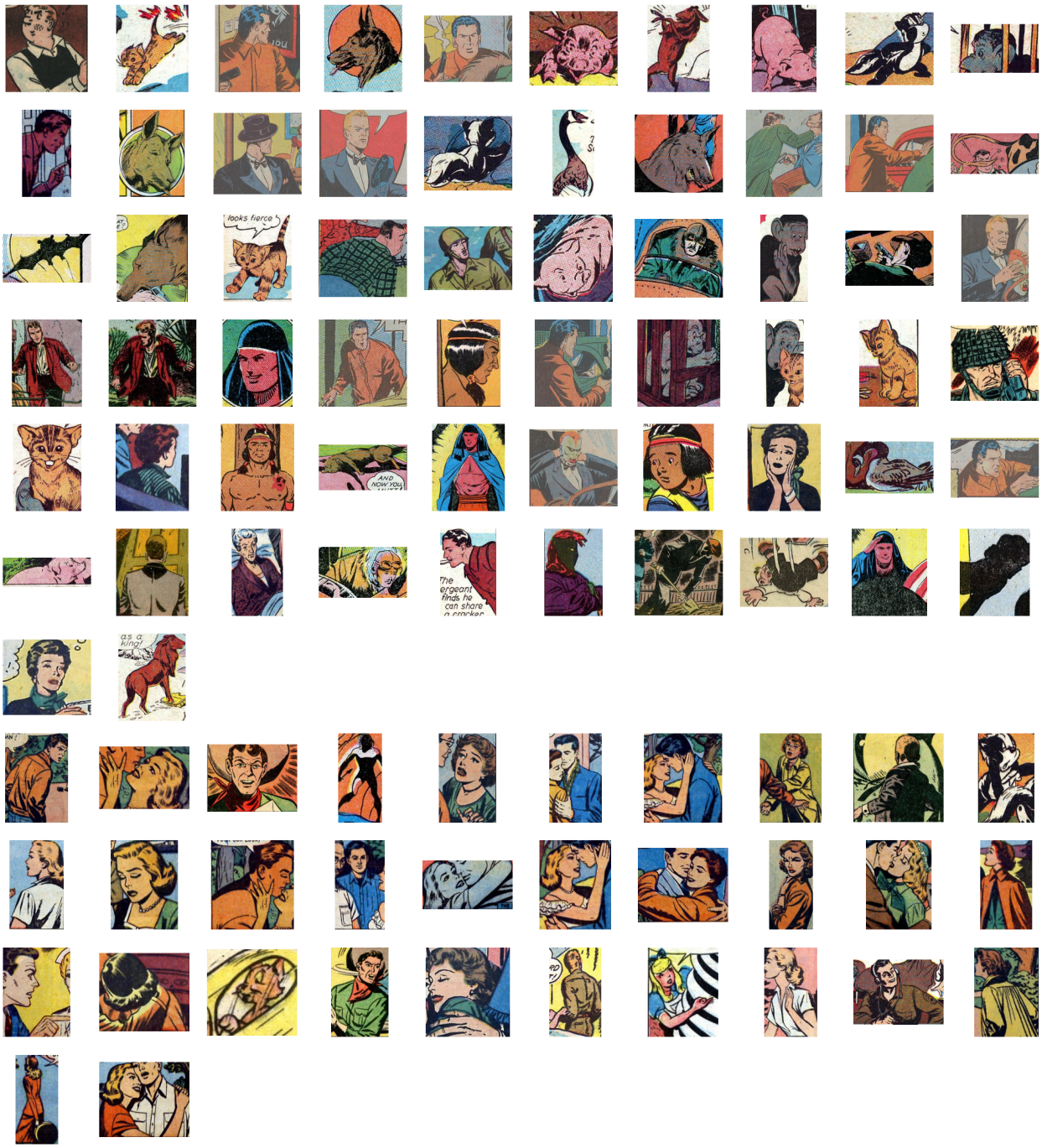

Figure 5.4: Clusters with confused characters

6. CONCLUSIONS AND FUTURE WORK

Based on our experiments, it becomes clear that zero-shot transformers GroundingDINO and SAM accomplished impressive results without any fine-tuning to comic-specific datasets. The most challenging dataset for them is the Manga images. This may be explained by the distinctiveness of these comics in terms of complex layout and the Japanese language. However, we consider that the performance would be improved after feeding transformers with comic images.

Thus, future work could be fine-tuning GroundingDINO and SAM in comic images. A good approach to creating a larger training dataset is to collect more comic images, use the GroundingDINO as a semi-supervised annotator, and then train them in a large amount of comic data. These models seem to have immense potential, but the process requires many computational resources and costs.

REFERENCES

[1] M. Azam, C. Sampieri, A. Ioppi, S. Africano, A. Vallin, D. Mocellin, M. Fragale, L. Guastini, S. Moccia, C. Piazza, L. Mattos, and G. Peretti, "Deep learning applied to white light and narrow band imaging videolaryngoscopy: Toward real□time laryngeal cancer detection," *The Laryngoscope*, vol. 132, 11 2021.

[2] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[3] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo: From yolov1 to yolov8 and beyond," *arXiv preprint arXiv:2304.00501*, 2023.

[4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[5] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[6] N.-V. Nguyen, C. Rigaud, and J.-C. Burie, "Comic mtl: optimized multi-task learning for comic book image analysis," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, pp. 265–284, 2019.

[7] T. Ogawa, A. Otsubo, R. Narita, Y. Matsui, T. Yamasaki, and K. Aizawa, "Object detection for comics using manga109 annotations," 2018.

[8] N.-V. Nguyen, C. Rigaud, and J.-C. Burie, *Multi-task Model for Comic Book Image Analysis: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part II*, 01 2019, pp. 637–649.

[9] R. Krusemark, "Comic books in the american college classroom: a study of student critical thinking," *Journal of Graphic Novels and Comics*, vol. 8, no. 1, pp. 59–78, 2017.

[10] H.-S. Kang, "Comic book project as a tool for teaching multimodal argument and fostering critical thinking skills: Implications for the l2 writing classroom," in *The College English Association Forum*, 2017.

[11] D. A. Yonanda, Y. Yuliati, and D. S. Saputra, "Development of problem-based comic book as learning media for improving primary school students' critical thinking ability." in *Elementary School Forum (Mimbar Sekolah Dasar)*, vol. 6, no. 3. ERIC, 2019, pp. 341–348.

[12] U.-R. Ko and H.-G. Cho, "Sickzil-machine: a deep learning based script text isolation system for comics translation," in *Document Analysis Systems: 14th IAPR International Workshop, DAS 2020, Wuhan, China, July 26–29, 2020, Proceedings 14*. Springer, 2020, pp. 413–425.

[13] N.-V. Nguyen, C. Rigaud, and J.-C. Burie, "Digital comics image indexing based on deep learning," *Journal of Imaging*, vol. 4, no. 7, p. 89, 2018.

[14] A. Kumar, S. Srivastava, and P. Chattopadhyay, "Machine and deep-learning techniques for image super-resolution," *Machine Learning Algorithms for Signal and Image Processing*, pp. 89–113, 2022.

[15] X. Yang, Z. Ma, L. Yu, Y. Cao, B. Yin, X. Wei, Q. Zhang, and R. W. Lau, "Automatic comic generation with stylistic multi-page layouts and emotion-driven text balloon generation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 2, pp. 1–19, 2021.

[16] C. Guérin, C. Rigaud, A. Mercier, F. Ammar-Boudjelal, K. Bertet, A. Bouju, J.-C. Burie, G. Louis, J.-M. Ogier, and A. Revel, "ebdtheque: a representative database of comics," in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 1145–1149.

[17] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, “Sketch-based manga retrieval using manga109 dataset,” *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017.

[18] K. Aizawa, A. Fujimoto, A. Otsubo, T. Ogawa, Y. Matsui, K. Tsubota, and H. Ikuta, “Building a manga dataset “manga109” with annotations for multimedia applications,” *IEEE MultiMedia*, vol. 27, no. 2, pp. 8–18, 2020.

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[20] C. Rigaud, “Segmentation and indexation of complex objects in comic book images,” *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, vol. 14, 12 2014.

[21] W. Sun, J.-C. Burie, J.-M. Ogier, and K. Kise, “Specific comic character detection using local feature matching,” in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 275–279.

[22] T.-N. Le, M. M. Luqman, J.-C. Burie, and J.-M. Ogier, “A comic retrieval system based on multilayer graph representation and graph mining,” in *Graph-Based Representations in Pattern Recognition: 10th IAPR-TC-15 International Workshop, GbRPR 2015, Beijing, China, May 13-15, 2015. Proceedings 10*. Springer, 2015, pp. 355–364.

[23] J. Lucas, A. J. Gallego, J. Calvo-Zaragoza, and J. C. Martinez-Sevilla, “Automatic detection of comic characters: An analysis of model robustness across domains,” in *International Conference on Document Analysis and Recognition*. Springer, 2023, pp. 151–162.

[24] F. S. Khan, R. M. Anwer, J. Van De Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, “Color attributes for object detection,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3306–3313.

[25] G. Soykan, D. Yuret, and T. M. Sezgin, “Identity-aware semi-supervised learning for comic character re-identification,” *arXiv preprint arXiv:2308.09096*, 2023.

[26] Z. Zhang, Z. Wang, and W. Hu, “Unsupervised manga character re-identification via face-body and spatial-temporal associated clustering,” 2022.

[27] C. Rigaud, J.-C. Burie, and J.-M. Ogier, “Text-independent speech balloon segmentation for comics and manga,” in *Graphic Recognition. Current Trends and Challenges: 11th International Workshop, GREC 2015, Nancy, France, August 22–23, 2015, Revised Selected Papers 11*. Springer, 2017, pp. 133–147.

[28] D. Dubray and J. Laubrock, “Deep cnn-based speech balloon detection and segmentation for comic books,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1237–1243.

[29] O. R. Vincent, O. Folorunso *et al.*, “A descriptive algorithm for sobel image edge detection,” in *Proceedings of informing science & IT education conference (InSITE)*, vol. 40, 2009, pp. 97–107.

[30] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[31] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[32] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[33] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu *et al.*, “Summary of chatgpt-related research and perspective towards the future of large language models,” *Meta-Radiology*, p. 100017, 2023.

[34] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," 2022.

[35] Ultralytics, "YOLOv5: A state-of-the-art real-time object detection system," https://docs.ultralytics.com, 2021.

[36] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun, "Objects365: A large-scale, high-quality dataset for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8430–8439.

[37] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, pp. 32–73, 2017.

[38] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 787–798.

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[40] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[41] L. H. Li*, P. Zhang*, H. Zhang*, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao, "Grounded language-image pre-training," in *CVPR*, 2022.

[42] C. Li, H. Liu, L. Li, P. Zhang, J. Aneja, J. Yang, P. Jin, H. Hu, Z. Liu, Y. J. Lee *et al.*, "Elevater: A benchmark and toolkit for evaluating language-augmented visual models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9287–9301, 2022.

[43] C. Rigaud, C. Guérin, D. Karatzas, J.-C. Burie, and J.-M. Ogier, "Knowledge-driven understanding of images in comic books," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, pp. 199–221, 2015.

[44] C. Rigaud, N. Tsopze, J.-C. Burie, and J.-M. Ogier, "Robust frame and text extraction from comic books," in *Graphics Recognition. New Trends and Challenges: 9th International Workshop, GREC 2011, Seoul, Korea, September 15-16, 2011, Revised Selected Papers*. Springer, 2013, pp. 129–138.