STATE-AWARE OBJECT UNDERSTANDING

Area Paper By

Nguyen (Will) Nguyen

Department of Computer Science

University of Rochester

April 2024

ABSTRACT

The advent of potent multimodal large language models alongside expansive datasets has markedly advanced visual understanding tasks. While the bulk of research in this domain has predominantly centered on scene understanding, the equally critical facet of computer vision, namely object understanding, has not garnered comparable attention from the academic sphere. Object understanding necessitates the capability of intelligent systems to fully grasp the attributes and conditions of objects, representing a core research challenge with extensive applications in domains such as human-robot collaboration and augmented/virtual reality (AR/VR). These applications range from progress estimation and defect detection to early problem identification. Nonetheless, much of the existing literature presupposes a static list of objects and states, framing the issue within the confines of classification or moment localization tasks. Such assumptions are misaligned with the dynamic nature of the real world, where gathering data on every conceivable object across various environments is impractical. Moreover, the textual content present on the surfaces of objects can offer valuable insights into their nature. This paper endeavors to address these underexplored questions within object understanding, particularly focusing on the comprehension of an object's state in an open-world context. Our approach begins with a thorough review of the literature on object understanding and state estimation, aiming to provide an overview of the current landscape in this field. Subsequently, we explore the role of text on objects in enhancing understanding and review the latest progress in scene text spotting. Additionally, we introduce preliminary efforts towards improving object state understanding and refining state-of-the-art scene text spotting techniques. We conclude by highlighting the limitations of existing works and discussing prospective challenges and directions for future research.

Human perception involves understanding both natural scenes and objects in the world. However, previous research has mainly focused on scene understanding, which extracts

knowledge from visual scenes while overlooking object understanding and the interactions between objects. This is a significant limitation because it's important to know the current state of an object, such as if it's ready to use or if it could cause any issues in the future. Focusing on tracking the attributes and states of objects is necessary in many types of videos such as egocentric and industrial chain surveillance cameras.

In this area paper, we explore the topic of object understanding, with a specific focus on comprehending the state of an object and recognizing text that might appear on its surface. We will begin by examining the importance of understanding the state of objects and their application in the real world. Additionally, we will delve into the role of scene text spotting in object understanding. Next, we will present our initial effort to tackle two fundamental problems: understanding the state of objects and improving scene text understanding models. For object state understanding, we use natural language to understand the state of the object. Besides, we use linguistic priors to improve the accuracy of the scene text spotting models. Finally, we will discuss our plan for future research on state-aware object understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Understanding how to interpret complex information about objects and their changing states is crucial for both human understanding and artificial intelligence systems. This ability has significantly influenced the development of specialized brain functions linked to our senses, like vision and hearing. Just as combining our senses enhances our perception, integrating different types of data, visual, textual, and contextual, is key to better understanding objects in various states.

Traditionally, computational models have focused on specific areas like understanding scenes or detecting objects in dynamic environments. These models usually ignore the state of objects, which lose a lot of information when observing the real world. For example, accurately detecting a defect or determining if an object is ready to use requires recognizing that an object's condition can change quickly.

Acknowledging these limitations, researchers are now working towards more adaptable and robust systems that can understand objects in more natural settings. This thesis explores how different attributes of objects interact with their states and uses a combination of methods to process this information similarly to how humans use their senses. We are looking into how advancements in natural language processing and computer vision can help us better understand objects in the real world, which could be useful in industries like manufacturing or in augmented reality systems.

The rest of this area paper is laid out to investigate the field of object understanding in depth as it applies to dynamic situations. We start with a thorough review of what's currently known about object state recognition and its applications in Chapter 2; we then discuss our initial research on understanding object states using natural language in Chapter 3; we show our effort to improve the scene text understanding models by leveraging lin-

guistic priors in Chapter 4; and we finish with a look at the challenges we face and where future research could go in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

2.1 Egocentric Video Understanding

Egocentric video understanding focuses on videos captured from the first-person perspective, which is quite different from the more common third-person viewpoint seen in most video datasets. This section discusses several key areas of research in this field, including the datasets used, how these videos help us understand interactions between people and objects, recognize activities, predict future actions, summarize video content, and analyze social interactions.

Egocentric Video Datasets: There are many special datasets created for studying videos taken from a person's point of view. For instance, EPIC-Kitchens [1] focuses on activities in the kitchen, while UT Ego [2] includes a wider range of daily activities, both inside and outside. The ADL dataset [3] captures everyday activities without specific instructions, unlike the Charades-Ego [4] and EGTEA [5] datasets, which involve more directed tasks in daily life settings. Ego4D [6] and Ego4D-EXO [7] are large-scale egocentric datasets that focus on a wide range of activities.

Human-Object Interactions and Activity Recognition: First-person videos are excellent for studying how people interact with objects because the camera gives a close-up view of these interactions. Many studies focus on detailed actions [8, 9] while some others [10, 11] have been conducted for recognizing diverse categories of activities from these videos.

Action Anticipation: Predicting what might happen next in a video is crucial for technologies that interact with or help people. Many of recent works [12, 13] focus on guessing upcoming activities, which is challenging due to the unpredictable nature of first-person videos.

Video Summarization: Summarizing videos involves picking out the most important parts from long recordings. Some previous works [14, 15] have developed methods to create short summaries of egocentric videos, which can be very varied and unstructured.

Parsing Social Interactions and Body Pose Estimation: Understanding social interactions and figuring out the body position of the person wearing the camera are also key aspects of studying first-person videos. Much research shows progress in analyzing social behavior and estimating poses from the wearer's perspective [16, 17].

2.2 Object State Understanding

2.2.1 Introduction

Understanding object states involves recognizing, monitoring, and predicting changes in objects over time. This area combines elements of computer vision, machine learning, and cognitive science. An object's state can include its physical features, position, operational condition, and interactions within its environment. Knowing an object's state is key for many uses in different fields. In industry, it helps in predictive maintenance by spotting parts that might fail soon [2]. For consumers, it improves how devices work based on how they are used [1]. In robotics, it aids in developing more independent systems that can better interact with their environment by handling objects according to their current state [3]. Various techniques have been created to identify and follow object states. These often use sensors and visual data, or both. Deep learning is effective at interpreting complex visual patterns to figure out object states. Traditionally, the combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are commonly used to process and analyze image or video sequences to spot state changes over time [8]. Besides, this problem can also be considered as the moment localization problem, which is localizing the frames of state/in progress/ end states [18].

Despite progress, several challenges remain. Objects can look different based on viewing angles, lighting, and obstructions, which can make it hard to accurately identify their

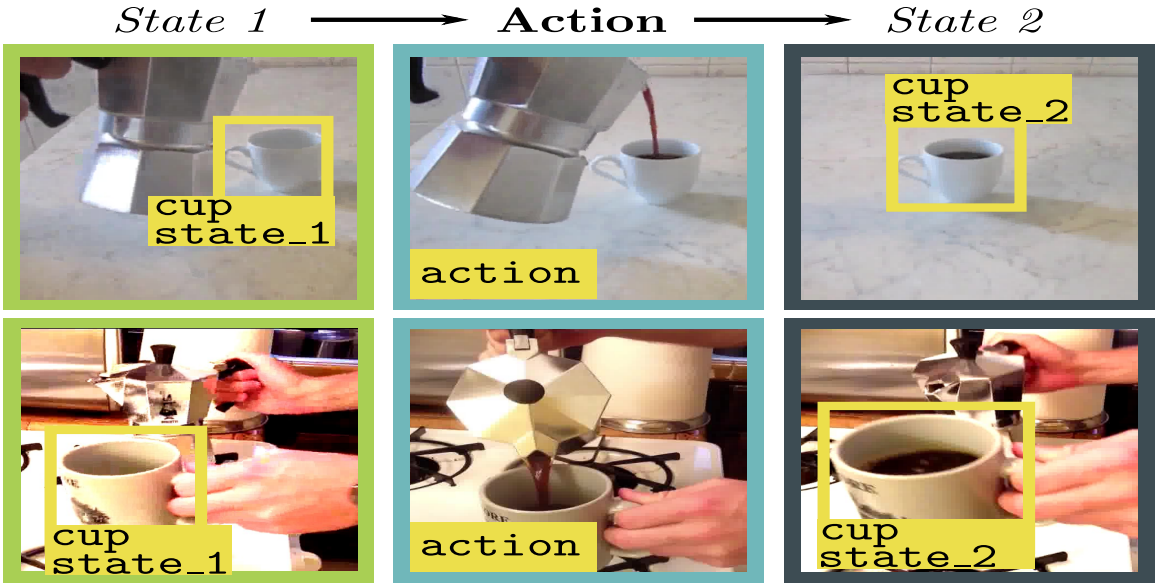


Figure 2.1: By observing people interacting with objects, the system can automatically detect object states, such as an empty or full coffee cup, and their corresponding manipulation actions.

state [9]. Another issue is the need for a lot of labeled training data, which is costly and hard to get, especially for rare or subtle state changes [10].

2.2.2 Related Works

In 2017, research on object states began to receive more attention, starting with articles that explored the relationship between manipulation actions and object states [19]. The initial idea is quite simple: to define an object's state change by having an action take place interacting with that object. Simply put, this study views the change in state of an object by a triplet in the order state 1 $\rightarrow$ action $\rightarrow$ state 2, as depicted in figure 2.1. In this example, the initial state of the cup is empty, after performing the action of pouring coffee into the cup, the subsequent state is that the cup is filled with coffee. Although the idea is simple, this research has laid the first foundations for the idea of exploiting the relationship between actions and changes in the state of objects. This sets the stage for many future studies in understanding object states as well as state changes.

However, this study still has many limitations because it is only the first effort in this

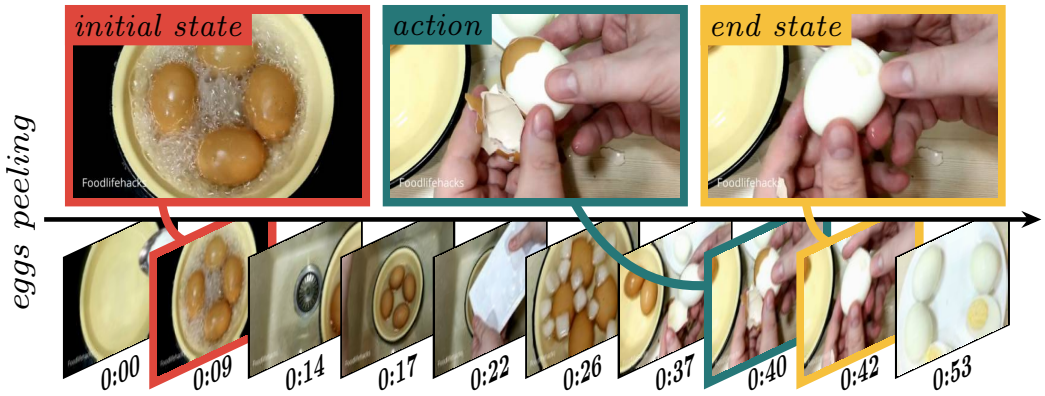


Figure 2.2: The figure shows an example of how a model learns about object states and state-modifying actions from a dataset of long, uncurated web videos. The figure displays video frames from the entire video, along with their corresponding timestamps. This figure highlights the challenge of identifying the precise temporal localization of the object states and actions in the entire video.

field of research. The first limitation that can be easily seen is that the research only focuses on a minimal number of object-action pairs. In the proposed data set, there are only seven pairs, including Put wheel, remove the wheel, fill a pot, open oyster, fill a coffee cup, open fridge, and close fridge. The second limitation is that this study requires human annotation to include a bounding box for the object to focus on and the location of frames state 1, action, state 2. This manual labeling requirement will require large costs. and especially not scalable, so it will not be possible to solve real problems when the model needs to be able to operate in the open world. The third limitation is when we need to perform complex requirements such as estimating how much time it will take for the model to complete, when this is especially important when tracking the progress of the work. In addition, simply localizing moments cannot help the model identify and support when people make mistakes.

By 2022, another study was published to solve the problem encountered in the above study about requiring human annotation that directly takes advantage of uncurated videos from the web [18]. This allows the model to take advantage of the huge amount of data from the web, learning knowledge from it at a very low cost in the data collection process. The way to formulate the problem is still similar to the previous research from 2017, which

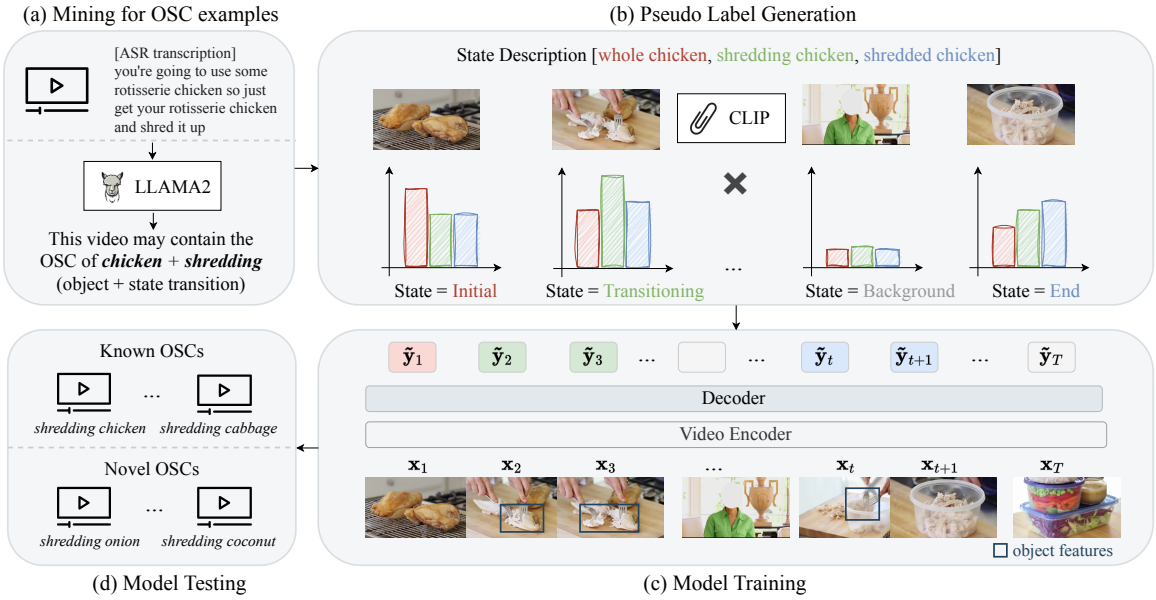


Figure 2.3: The VIDOSC framework is outlined as follows: (a) Mining for OSC examples: ASR transcriptions and videos, with LLM capabilities, are used to automatically find OSC examples. (b) Pseudo Label Generation: Textual descriptions and a VLM are used to create supervisory signals for training. (c) Model Training: A video model is developed to predict states without needing to know the object type. (d) Model Testing: The model is tested in a new way, checking how it performs on both familiar and new OSCs. Notably, even though text helps during training, the model relies only on video data during tests, ensuring it is very flexible and practical. The ground truth for the test set is manually annotated.

is to predict the triplet initial state $\rightarrow$ action $\rightarrow$ final state. However, the difference here is that the training process does not require labels but relies on guidance signals about the order of the three groups of frames mentioned above. Research also shows that if order information is well utilized, this is also an effective signal for understanding object status.

However, this approach still has certain limitations. The first limitation is that this method requires us to train separate models for different action-object pairs, meaning each will require a separate model. The second problem is that the approach still relies on localizing a set of three frames from the video, which limits the ability to understand complex states and track the progress of objects and actions. The final problem lies in the fact that, based on the design of tasks and models, this approach still cannot solve the open-world problem and is difficult to apply in practice.

To solve the problem of understanding open-world object state change, the VIDOSC

framework [20] is proposed to classify frames into three types of states: initial, transitioning, and end. In addition, frames without objects will be assigned as background frames. The figure 2.3 shows the procedure of VIDOSC framework. First, the author filters out videos with object state changes using ASR transcripts and then uses these transcripts to ask LLAMA 2 to select satisfactory videos with object state changes. These videos will then be labeled for each frame by comparing each frame with three specific states predetermined using the CLIP model. The state with the highest CLIP score will be assigned as the label for that frame. Frames with almost equal and low CLIP scores will be considered background frames and have no objects in them. After this automatic labeling process, a model will be trained using the data synthesized above. During testing, the model is tested with both known objects and unseen objects.

The VIDOSC framework has some limitations that present significant challenges in its ability to understand object state changes in videos. Firstly, the framework's classification system is limited to only three types of states, which could cause it to miss more complex or gradual transitions that do not fit neatly into the initial, transitioning, and end categories. Real-world scenarios often require recognizing a broader spectrum of states for more detailed analysis. Secondly, the framework's "open-world" capability is limited because it relies on known actions, which means it cannot accommodate actions it has not been trained on. This limitation reduces its usefulness in dynamic real-world situations where unpredictability is the norm. Lastly, the framework lacks interactive capabilities such as QA or conversation, which makes it unsuitable for scenarios where context from human interaction or additional clarification is needed. For instance, in user-oriented applications where conversational AI can provide valuable insights or guidance based on video content analysis, VIDOSC's current structure falls short. The inability to support a dialogue means that users cannot probe deeper into the analysis, ask for explanations, or refine the results through feedback. This limitation reduces the framework's potential in user-centric environments where interactivity is key to functionality and user experience.

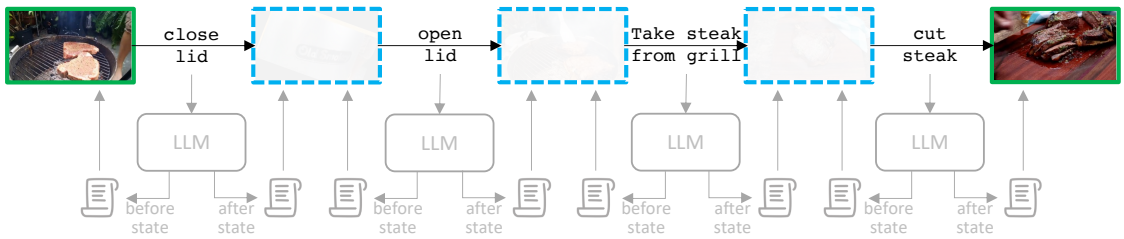


Figure 2.4: Each step is represented as a state change, with descriptions generated by LLMs utilized for state representation learning.

SCHEMA [21] does not focus on object state understanding but uses object state as a guidance signal to help improve the performance of the planning procedure model. In this article, the object state change is used as important information directly related to a sequence of actions that lead to changing the state of an object from the initial state to the goal state, as described in figure 2.4. The paper's idea is to take advantage of LLM's reasoning capabilities to generate the state of the object before and after performing the action. Once the state of the object is represented in text, this information will be used in both steps of the model: mid-state generation and procedure generation. The results show that the performance of procedure planning increases significantly when information about object state changes is incorporated. It is noteworthy to mention that SCHEMA and our preliminary initiative, OSCaR, are concurrent developments in the field. Both works progress in parallel, each contributing unique insights and techniques to the domain of object state understanding.

The limitations of the SCHEMA model stem from its approach to state representation and interaction capabilities. The model's reliance on the LLM's reasoning for state representation without incorporating visual input mirrors the limitations of "blind" models. This reliance indicates a potential oversight of the rich contextual information that visual data could provide, possibly affecting the model's robustness and accuracy in real-world scenarios where visual cues are pivotal. Moreover, SCHEMA's inability to engage in QA or conversational interactions limits its functionality in user-interactive settings. Without the capacity for dialogue, SCHEMA cannot clarify ambiguities, answer questions, or refine

its performance based on user feedback. This lack of interactivity could hinder its deployment in applications where human support is essential, such as instructional guidance or collaborative robotics.

2.3 Scene Text Spotting

2.3.1 Introduction

Text detection and recognition in natural scenes, commonly referred to as scene text spotting, represents a critical area of research with profound implications across various domains. Applications of this technology are vast and impactful, including assisting visually impaired individuals [22], enhancing robotic navigation [23], and facilitating mapping and localization efforts [24]. The challenge in scene text spotting arises from the inherent ambiguity presented by texts in natural environments due to factors like aesthetic variations, environmental degradation, and poor lighting conditions. While the use of lexicons or dictionaries can mitigate such ambiguities to some extent [25], it often complicates the underlying models, increasing their complexity.

A typical end-to-end scene text spotting framework encompasses two principal components: text detection and text recognition. Over the years, there has been significant progress in both areas [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]. However, despite these advancements, the performance of current scene text spotting models still lags behind human capabilities, particularly in recognizing distorted or blurred text. This gap is primarily due to the sophisticated use of linguistic knowledge in human reading, which aids in disambiguating text more effectively. Existing models, particularly autoregressive text recognition models [39, 40, 41, 42], attempt to harness linguistic structures but fall short of fully leveraging linguistic knowledge due to the limited scope of scene text datasets and the simplistic one-hot vector encoding employed in these models.

The challenges do not end at recognition. During inference, text detection models often misinterpret the spatial relationships between characters or words, leading to errors in text

detection and recognition. Traditional methods employing one-hot labels have shown to enhance learning in detection models when recognition constraints are considered. However, they fail to account for the contextual relationship among characters within a word. By incorporating linguistic priors based on the structural knowledge of how words are typically formed and spaced, detection models can improve in both accuracy and efficiency.

2.3.2 Related Works

Text Recognition Approaches in Scene Text Spotting: Text recognition within the field of scene text spotting can be categorized into two primary methodologies. The first approach is character segmentation and recognition, where a text region is divided into individual characters that are recognized separately [43, 44, 45, 46, 47]. This method, while direct, often fails to capture the contextual relationships between characters, leading to potential inaccuracies in character recognition when processed independently.

The second and more advanced approach utilizes auto-regressive models, particularly those based on recurrent neural networks (RNNs) integrated with attention mechanisms or CTC loss, to decode text sequences [48, 49, 50, 51, 52]. Auto-regressive models, such as CRNN, employ bidirectional LSTMs to predict the probability distribution of text sequences, effectively reducing character duplications through CTC loss [53, 54, 39, 55, 40, 41, 42]. Innovations like SCATTER and Charnet further enhance the robustness of these models by integrating attention mechanisms that fuse character-level with word-level encodings, thereby improving the accuracy of text recognition [56, 57].

This sequential processing enables the implicit integration of language model dynamics, akin to probabilistic language models used in natural language processing [58, 59, 60]. Despite their sophistication, these RNN-based methods are not without limitations, primarily due to the restricted vocabulary often present in the training datasets, which hampers their ability to fully leverage comprehensive language models. This gap in potential underscores the ongoing challenge within scene text recognition to develop methods that can

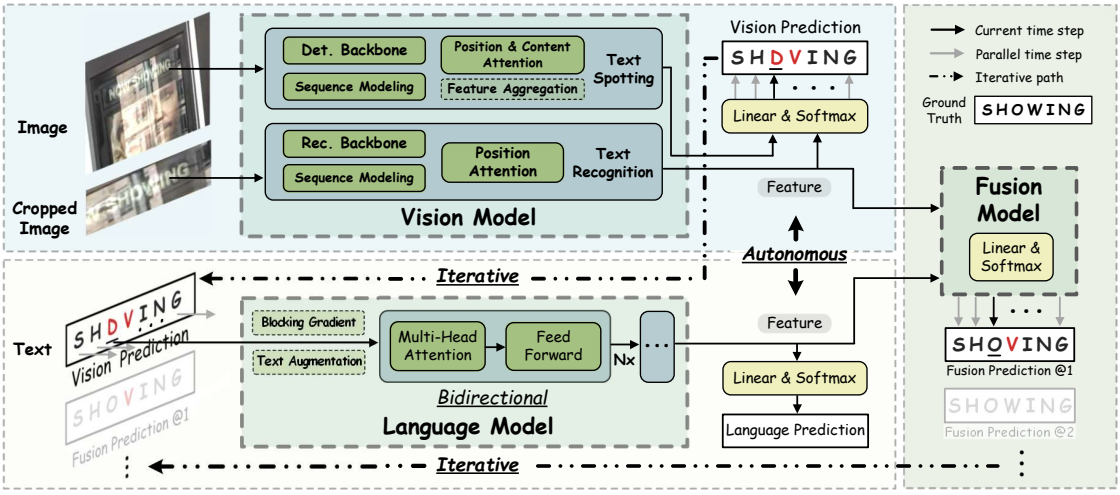


Figure 2.5: ABINet++ pipeline.

seamlessly integrate robust language models while accurately interpreting and processing textual content in diverse settings.

Language-Driven Scene Text Recognition: The integration of linguistic knowledge into scene text recognition models is gaining traction [34, 61, 62, 63]. VisionLAN and ABINet, for instance, incorporate visual and linguistic reasoning, enhancing recognition accuracy through iterative processing and language model utilization. Despite their effectiveness, these models often suffer from increased complexity and computational demands, especially in settings requiring iterative processes.

Knowledge Distillation: Knowledge distillation offers a pathway to enhance model performance through the transfer of knowledge from a more complex ”teacher” model to a simpler ”student” model [64]. DeiT distinguishes between soft and hard-label distillation approaches, where the former involves using the teacher's output probabilities directly, and the latter converts these probabilities to one-hot encoding. This methodology is particularly advantageous for transferring complex knowledge without significantly increasing the computational burden on the student model.

Although incorporating linguistic priors into scene text recognition has become a popular trend in scene text recognition, exploiting this knowledge for scene text spotting has

not been studied much. ABINet++ [65] is an extension of ABINet for scene text spotting that is the first attempt at integrating linguistic knowledge for scene text spotting. Although achieving good results in terms of results, ABINet++ still has two major limitations. The first limitation is that ABINet uses a pretrained language model as an external knowledge base and requires repeated interactions with this language model during the run to refine the results. Repeated running makes the results more accurate, but also creates time complexity overhead, making it difficult to run in real time. The second drawback is that although the results show that linguistic prior not only increases the performance of the recognition step but also improves the text detection step, ABINet++ does not provide analysis or a reasonable explanation for this improvement. From there, developing a method that can integrate linguistic knowledge into scene text spotting models without increasing the complexity of the models, and providing an explanation for how linguistic priors help scene text detection are two important issues need to be resolved.

2.3.3 Why Scene Text Understanding is Important for Object Understanding?

In many real-world scenarios, objects are often inscribed with text that provides essential information about their state and usage. The ability to read and interpret this text is crucial for informed decision-making. For example, determining the suitability of a soft drink for consumption requires checking the production and expiration dates printed on the can. Similarly, safe medication usage mandates reading dosage instructions detailed on the packaging. Optical Character Recognition (OCR) has long been established as a fundamental task in computer vision. However, it remains a formidable challenge, particularly for languages other than English. The significance of OCR extends beyond mere text recognition; it plays a pivotal role in the broader context of object understanding. Enhancing the performance of OCR models is not only crucial for improving text recognition capabilities but also significantly contributes to the advancement of object understanding research. This dual utility underscores the importance of ongoing research efforts aimed at refining OCR technologies.

CHAPTER 3

OBJECT STATE CAPTIONING AND STATE CHANGE REPRESENTATION

3.1 Introduction

The field of Natural Language Processing (NLP) has evolved beyond mere text interpretation and generation, advancing into realms where understanding and interacting with the physical world becomes imperative. From studying causal reasoning [66] to building a

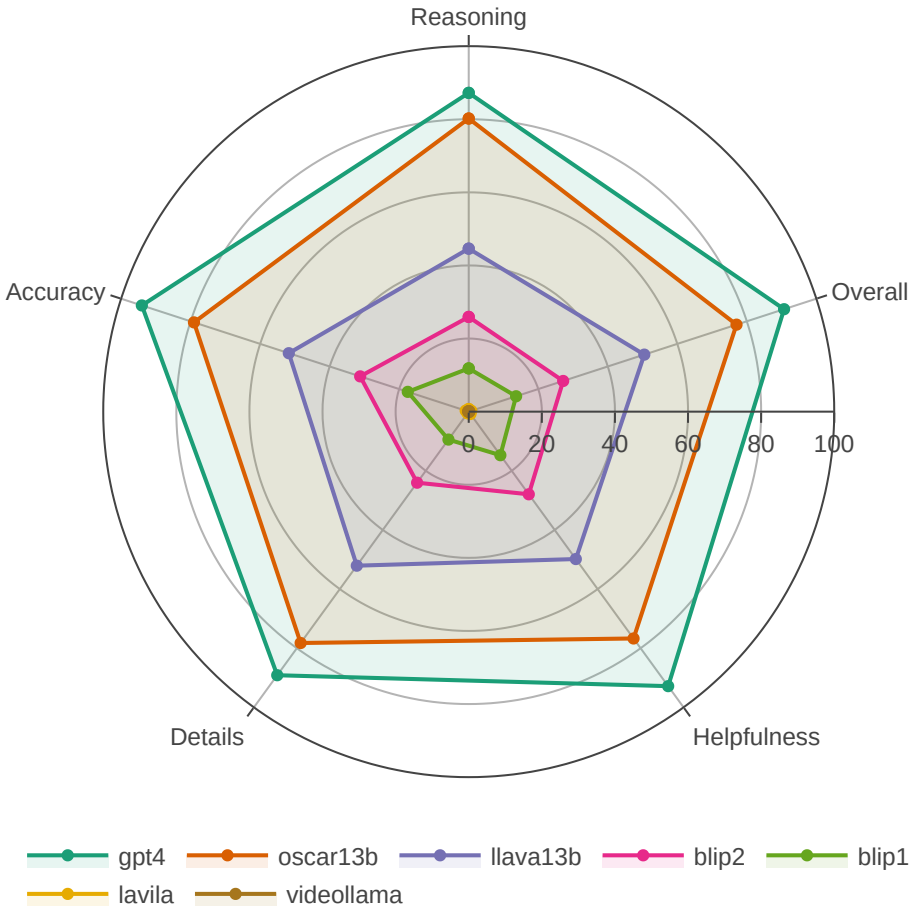


Figure 3.1: **Surpassing prior models in aligning with human judgements.** Our method achieves near parity with GPT-4V ratings across helpfulness, accuracy, reasoning, and other key metrics.

world model for cause-effect prediction [67, 19], researchers have been working on the

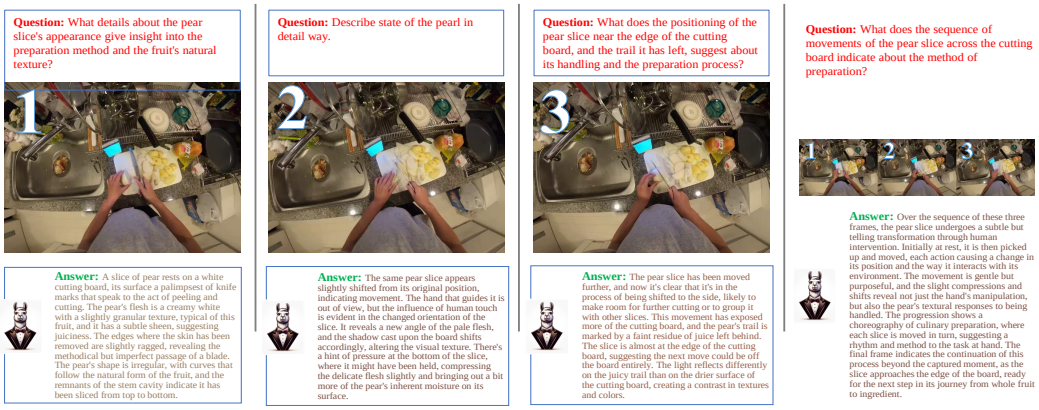


Figure 3.2: **OSCaR's description of state, state change, and illustration of reasoning.** State description involves the characterization of a specific region of interest within the video and the associated activity. State change entails the description of the evolution of a system over a defined temporal sequence. Furthermore, the analysis of the state of an object is centered on comprehending and elucidating the mechanisms underlying the object's evolution.

problem of causation in the physical world.

In this paper, we investigate the very basic causal relations between a concrete action and the change of the object state caused by this action. For example, given an image as shown in Figure 3.2, we, as humans, would have no problem understanding which object is being actively interacted with. Furthermore, given the statement "cutting the bread", we would naturally imagine what state change may happen. However, Despite tremendous progress in knowledge representation, automated reasoning, and machine learning, artificial agents still lack the understanding of naive causal relations regarding the physical world [66].

Imagining a scenario where artificial agents collaborate with humans in the physical world, they will need to understand the physical action effect to reason, learn, and assist humans [68]. To empower machines with such capabilities, this paper introduces a novel benchmark focusing on understanding object state changes from egocentric visual inputs, which has the advantage of the lens of human eyes.

Understanding object state change is not only a complex task but also practical and foundational for many other tasks, such as helping intelligent agents to understand the envi-

ronment dynamics and complete task [69, 70, 71], tracking the state of dialog[72], creating causal graphs for knowledge representation for complex question and answering [73].

Modeling object state change requires two abilities: 1) scene understanding, which involves parsing the world through an object-centric lens, and 2) causal-effect understanding, which entails identifying likely actions and their effects by observing images before, during, and after an action.

Previous research efforts have concentrated on building symbolic representations to ground changes and states [74, 75, 76]. However, given the diversity and complexity of objects and their states, influenced by contextual and temporal factors, symbolic representation alone falls short. This paper proposes the use of natural language as a more expressive and intuitive medium for this task. This approach not only aligns the understanding of visual content between humans and AI systems but also enhances communication between them, providing a richer context than unimodal models.

Essentially, we form the scene understanding as an object-centric visual captioning problem. We can utilize natural language to describe the objects and any changes that may occur. On the other hand, the ability to understand the causal effect is formed as a visual question-answering problem based on 3 images: before, during, and after the action. Our dataset and experiments exhibit considerable potential for scalable application across various domains in future research. While conducting this study, another research was also conducted to understand object state change with a different approach [77]. That shows the importance and significant potential of this research direction.

In summary, our contributions are threefold:

- We introduce a new problem to understand states and state changes of object through natural language.
- We present a method to generate good-quality visual instructions guided by simple annotations, applicable to both images and videos, advancing future research in visual instruction tuning. Our pipeline provides a good starting point for the data

collection process.

- Our paper introduces OSCaR, a novel dataset and a benchmark leveraged by the power of GPT-4V that contains different tasks for object state understanding, including visual captioning, visual question answering visual dialog, and reasoning.

3.2 Related Works

Object state change: Localizing and recognizing changes of object states, play a key role in applications such as procedural planning [78], robotics, and video action understanding [79, 80, 81, 82, 83]. Recognizing object state changes necessitates the joint discovery of states and actions through an understanding of their causal relationship, as discussed in prior works [19, 84, 18, 85]. Recently, a self-supervised method has been proposed to jointly localize action and state changes temporally from noisy untrimmed long videos [18]. Moreover, [86] introduces a novel benchmark for the generation of object states, yet their focus is very limited to only the cutting action and a small dataset. However, previous studies often separate scene understanding from object state change recognition and tend to operate under a closed-world assumption, which limits their applicability in real-world scenarios. Our research aims to bridge the gap between human and machine perception by integrating egocentric views and language.

Multimodal Large Language Models: Recent advancements in Large Language Models (LLMs) [87, 88, 89, 90] have led to significant achievements in language understanding and generation. This progress has sparked an interest in the creation of MLLMs that blend the advanced linguistic processing of LLMs with capabilities for multi-modal perception [91, 92, 93, 94, 95, 96]. The core of this research is the fusion of pre-training visual encoder representations with the input embedding space of LLMs, achieved by pretraining with datasets that interleave images and text. [97, 98, 99]. In this paper, we aim to provide a comprehensive evaluation of these models, particularly focusing on their performance in object state change recognition.

3.3 The OSCaR Dataset

This section outlines our pipeline for creating visual instructions on object states. We begin with the process of collecting diverse visual data from public sources, detailed in section 3.3.1. Following this, section 3.3.2 describes our approach to enhancing data quality using simple human annotations across various tasks, facilitating a deeper understanding of object states. Our method enables the generation of detailed captions, visual question answering, and visual dialogue.

3.3.1 Video Collections

OSCaR is a curated compilation of videos sourced from two distinct datasets: EPIC-KITCHENS [1] and Ego4D [6]. Acknowledging that changes in object states occur progressively over time rather than abruptly within a single frame, we have selectively included video clips that effectively illustrate these state transitions. Our selection process ensures that these videos depict the dynamic changes in object states and capture moments where the objects remain stationary for short enough durations. This approach enabled us to compile a comprehensive visual dataset encompassing the object's static and transitional states.

We initially analyzed the verbs from the original videos of the EPIC-KITCHENS dataset to ensure that the videos highlighted objects undergoing state changes. We categorized these verbs into three groups: *change*, *not sure*, and *not change*. The *change* group consists of verbs likely to alter the state of objects, including actions like Open, Close, Wash, Cut, and Mix. Conversely, the *not change* group encompasses verbs with a minimal likelihood of inducing state changes, such as Take, Put, Move, Check, etc. Lastly, the *not sure* group includes verbs with ambiguous potential for state change, covering actions like Shake, Flip, Use, Pull, and others. After filtering the EPIC-KITCHENS dataset, we were able to identify 69 verb classes that consisted of a total of 650 verbs. Using this verb list, we retrieved all video segments containing those actions.

Upon analyzing the videos, we discovered that some objects only appeared in a few times. As a result, we split the videos into two groups. The first group comprises videos that focus on objects that occurred more than ten times, and it will be used to construct our training and testing set. The second group includes videos with objects that occurred less than ten times. These objects are rare in EPIC-KITCHENS and can be used for open-world evaluation, which will be discussed in section 3.4.2. In the first group, we randomly selected 10 to 50 video segments per object, resulting in 7442 with 306 different objects from EPIC-KITCHENS.

We leveraged Ego4D, the largest egocentric video dataset, selecting video segments tagged with "$object\_of\_change$" to enhance our data's diversity. This tag highlighted videos showcasing object state changes. By gathering these specific videos, along with details of the objects and their narrations, we informed our data generation and compiled relevant statistics. From this dataset, we extracted 5942 segments featuring 296 unique objects for our OSCaR project.

3.3.2 GPT-assisted Data Generation

Caption Generation: Captioning plays an important role in visual understanding. Understanding object states requires detailed and informative captions to capture the exact state of objects. To achieve this goal, we generated captions for all collected videos by leveraging GPT-4V and human's weak annotations. This problem requires two types of annotations, including 1) Start and end frame ID in videos during the event to make state changes and 2) A short description of what happens in the video. The short description can be a verb representing the action and a noun representing the object humans interact with (e.g., washing tray). We designed adaptive prompts to inject this annotation as context to guide GPT-4V to generate high-quality captions. We found that GPT-4V often suffers from ambiguity without this guidance, and the quality of generated captions is degraded. With simple human guidance, GPT-4V can reduce ambiguity and produce better-quality

captions.

Multiple-choice QA Generation: The multiple-choice question is a method of presenting a set of answers, including incorrect options, to teach machine learning models how to distinguish between correct and incorrect answers. This type of question can also be used as a form of instruction, where the question serves as the prompt, and the answer serves as the response for the models. We created multiple-choice question and answer sets based on generated captions.

Conversation Generation: Visual dialog is a complex task requiring understanding of visual content and conversation context, and faces challenges in data collection due to its need for natural dialogues between two people viewing the same content. This process is time-consuming and resource-intensive, especially when involving reasoning and explanations. With the growth of machine learning models, generating visual dialog data is increasingly vital. We've developed a method that uses captions to create visual conversation data, enhanced by GPT-4V's ability to provide explanations, offering flexible and diverse data. This approach, labeling input data for images and videos, is cost-effective and faster than manual methods, generating vast amounts of training data for future models.

3.4 OSCaR Benchmarks

3.4.1 Evaluation with Text Generation Metrics

The dataset we are providing consists of 500 videos from the Ego4D and EPIC-KITCHENS datasets, which are specifically designed for benchmarking purposes. Each video is annotated by four detailed captions, all of which have undergone rigorous human verification to ensure the quality and reliability of this evaluation set. To ensure a comprehensive and accurate assessment of performance, text generation metrics such as BLEU, Rouge, LSA, among others, can be used for evaluation purposes.

3.4.2 Open-world Object State Understanding

Collecting data for all objects worldwide and then training models is not feasible. However, humans can describe new or unfamiliar objects, which can be challenging for AI, especially when they are in a new domain or serve a different purpose. Fortunately, recent achievements in MLLMs have opened up the potential for AI to have this ability. During pre-training with large amounts of data, MLLMs can learn general knowledge about the world. Besides, models will learn how to perform tasks during the visual instruction tuning process. In both processes, the models may or may not have been exposed to objects not in the object state understanding training set. The question is whether models can generalize to objects of this type. To answer this question, we provide two evaluation sets to test the generalizability of the models.

Cooking domain objects have not occurred in the training set for object state understanding: For this evaluation, we want to investigate the model's ability to understand objects that have not appeared in the training set in a similar scenario with the training domain. We provided a set of 2,485 videos with 1,024 objects that have not occurred in the object state training set. This testing set will evaluate how in-domain knowledge can help models understand object states and state changes. We used GPT-4V to annotate 344 videos for evaluation purposes.

Out-of-domain objects state understanding: This evaluation focuses on judging the ability of models to understand objects beyond the training domains. Our training set contains only the cooking domain data, while this testing set has diverse domains, such as baker, household management, cleaning/laundry, bike mechanic, etc. This set was extracted from the Ego4D dataset and contains 43,367 videos with more than 500 objects. This testing not only can be used for evaluation but also has the potential to scale up using our pipeline for object state understanding in other specific domains. For this evaluation set, we selected 10 videos from each of the 51 different domains, totaling 356 videos. Domains with fewer than 10 videos have all their videos included. This set is also annotated by GPT-4V.

3.4.3 Data Quality Verification

We evaluated the quality of descriptions for object states and activities across video frames using Amazon MTurk for human feedback. Our assessment framework included five guidelines for spotting inaccuracies, focusing on frame-specific description accuracy, two for assessing state change accuracy, two for identifying hallucinations, and three for recognizing incomplete descriptions. Annotators were asked to categorize each description under one of four labels: 1) Fully Detailed and Comprehensive, 2) Generally Complete with Minor Omissions, 3) Lacks Important Details or Contains Errors, or 4) Incomplete, Misleading, or Hallucinating, and provide reasoning to discourage random responses. This study utilized 500 samples from the EPIC-KITCHENS and Ego4D datasets, leading to the validation of 2000 natural language descriptions.

3.5 Data Statistics

In order to help models generate concise and informative answers, we have defined short answers as those with less than ten words and long answers as those with more than ten words. Short answers provide brevity, while long answers offer detailed and informative information. The distribution of these two types of answers can be seen in Figure 3.3. The average answer length in the dataset is 47.06 words. Long answers make up about 75% of the data, with an average length of 63 words, while short answers account for about 25% of the data, with an average length of 3.32 words. By splitting the data accordingly, future models can provide short, direct, and informative answers with explanations. To showcase the uniqueness of our OSCaR dataset, we have presented a comparison between OSCaR and other related datasets in Table 3.1. The OSCaR dataset comprises a vast number of instructions, along with images and videos. Additionally, it also provides data for object state captioning and object state change captioning.

In section 3.4.2, we discussed two types of open-world datasets for object state under-

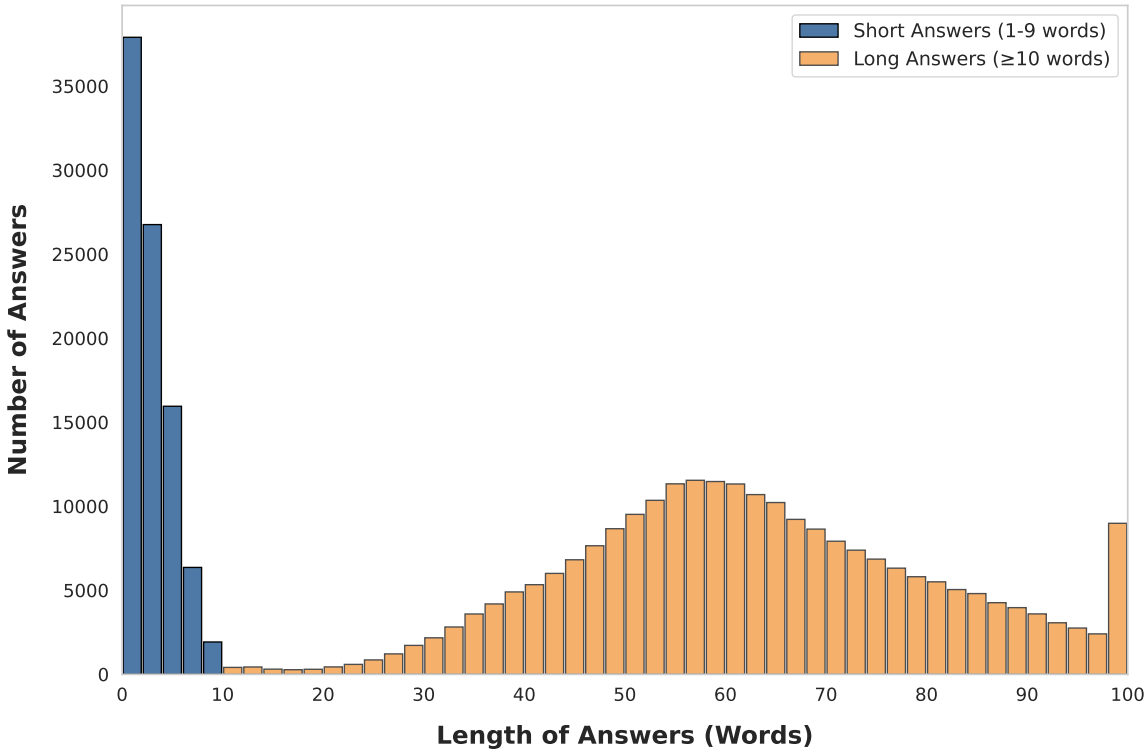


Figure 3.3: **Distribution of answer lengths.** The figure shows how answers are distributed by length in the dataset. It separates short answers (1-9 words) from long answers ($\geq$ 10 words). The histogram displays the number of answers on the y-axis based on increasing answer lengths on the x-axis. There is a category at 100 words for answers with lengths greater than or equal to 100 words. This breakdown emphasizes the balance between brief, direct answers and more detailed, explanatory responses.

standing: in-domain cooking and open domains. Although we trained on videos with object state changes, in open-world evaluation, we tested the models on both types of videos, with and without object state changes, to ensure their generalizability. The in-domain evaluation set consists of 2,485 videos with 1,024 novel objects extracted from EPIC-KITCHENS.

We have extracted an open-domain evaluation set from the Ego4D dataset. The top 10 most frequent domains in the open-world testing set are shown in Figure 3.4. This evaluation set from 51 different domains, contains annotations for domain, action, object name, and action narrations extracted from annotations of Ego4D. The set includes 43,367 open-world videos for which we know their domains and 56,231 videos of unknown domains, but we still have information about their object names and action narrations. Thus, this set

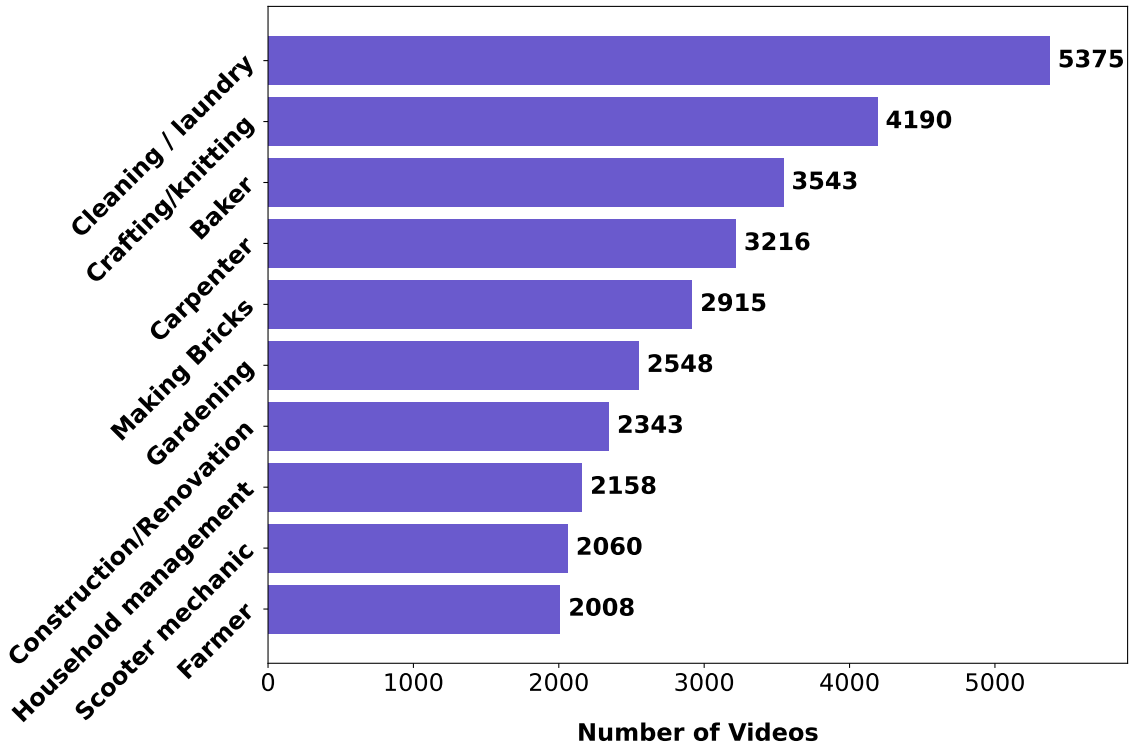


Figure 3.4: **Top 10 open-world domains (excluding cooking)**. The figure shows non-cooking domains present in the open-world test set used to assess model generalization. By evaluating performance on household and occupational activities unseen during training, we benchmark the trained models' capacity to understand new objects and actions beyond cooking tasks.

can be utilized not only for open-world evaluation but also for the advancement of general domain object-state understanding in the future when applying our method to generate labels. This set of data hasn't been annotated, but the data we extracted from Ego4D are ready to use our pipeline to scale up the data generation.

3.6 Experiments

In this section, we will discuss the experimental design we used and how we trained our model. Our fine-tuning process will be described in Section 3.6.1. Additionally, we included other vision language models such as BLIP [104] , BLIP2 [97], LaViLa [105], and Video-LLaMA [106] for comparison purposes. Firstly, we will evaluate our model's performance in the cooking domain in Section 3.6.3. After that, we will also evaluate its

Table 3.1: Comparison of OSCaR dataset versus other related datasets. OSC and OSCC represented for Object State Captioning and Object State Change Captioning, respectively.

Dataset	Video	#Clip	#Instruction	OSC	OSCC
MiniGPT-4 [100]	✗	✗	5K	✗	✗
Shikra-RD [101]	✗	✗	5.9K	✗	✗
LLaVA [102]	✗	✗	345K	✗	✗
VideoChat [103]	✓	11K	20.8K	✗	✗
OSCaR	✓	**18K**	**400K**	✓	✓

performance in an open-world setting in Section 3.6.4.

3.6.1 Model Training

We conducted extensive experiments to showcase the effectiveness of our data generation pipeline in solving object-state understanding problems. A straightforward approach to solving these types of problems is using a model with a text encoder to encode prompts and a visual encoder to encode visual content. After that, both of these inputs will be used as conditions to generate text answers with a text decoder. Ideally, this text decoder will be an LLM.

We fine-tuned LLaVA, an open-source MLLM featuring capabilities like visual dialogue, question-answering [107], and OCR [25, 108], to achieve our goals. Notably, the generated data can enhance any future vision-language models beyond LLaVA. We experimented with LLaVA using Vicuna 7B and 13B models under two conditions: with and without its original visual instruction tuning data, referring to the former as OSCaR.

For training, we employed Lora fine-tuning with a configuration of rank 128 and alpha 256, using Vicuna 13B and 7B models alongside the OpenAI/CLIP-ViT-Large-Patch14-336 vision encoder. A projector transformed visual features into tokens. Our fine-tuning parameters included a single epoch, a learning rate of 2e-4, a batch size of 16 per device, and a maximum model length of 2048.

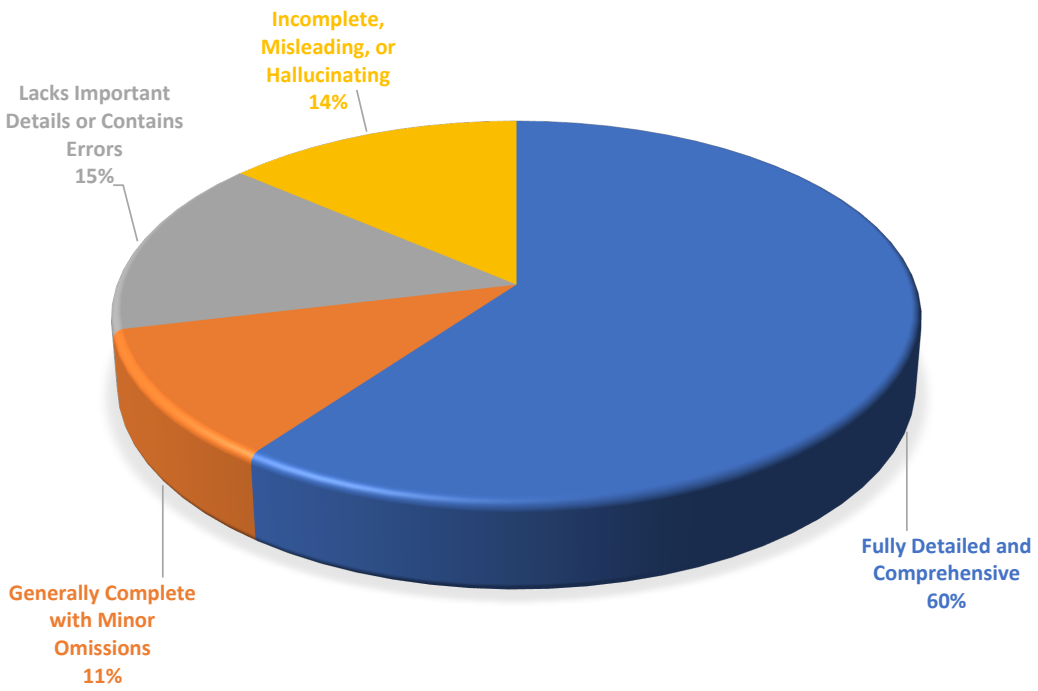


Figure 3.5: **GPT-4V zero-shot caption quality human evaluation.** The figure shows the distribution of quality ratings assigned by human annotators evaluating frame descriptions automatically generated by the GPT-4V model under zero-shot conditions. Descriptions for 500 video frames were rated.

3.6.2 Evaluating GPT-4V

Because our pipeline uses GPT-4V as the knowledge model to annotate our data, evaluating GPT-4V's ability is crucial. Evaluating GPT-4V's performance has two purposes: 1) Understanding the performance of GPT-4V on this task and 2) Producing a clean benchmark beyond the ability of GPT-4V for future research. As discussed in section 3.4.3, we ask humans to check data quality and classify quality into four levels with text explanation. Figure 3.5 shows the distribution of data quality from 500 videos sampled from the dataset for benchmarking.

3.6.3 Evaluation on Cooking Domain Objects

Text Generation Metrics Evaluation: The table 3.2 in this document displays the results of two text generation metrics, BLEU and ROUGE. As per the table, LaViLa and BLIP1

Table 3.2: **Performance comparison based on BLEU and ROUGE scores.** OSCaR is LLaVA fine-tuned with OSCaR data, mixed data is a combination of LLaVA data and OSCaR data.

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
LaViLa [105]	0.006	3.3	0.26	3.27
BLIP1 [109]	0.008	1.38	0.08	1.35
BLIP2 [97]	0.1	11.53	2.12	10.51
Video-LLaMA [106]	1.0	17.75	2.69	16.02
LLaVA v1.5 13B [102]	3.72	27.09	6.59	24.01
LLaVA v1.5 7B [102]	3.23	25.37	6.22	22.60
OSCaR 13B (OSCaR data only) (Ours)	5.28	27.93	7.67	24.45
OSCaR 7B (OSCaR data only) (Ours)	5.1	28.27	7.42	24.77
OSCaR 13B (Mixed data) (Ours)	5.76	29.26	8.24	25.78
OSCaR 7B (Mixed data) (Ours)	**5.79**	**29.94**	**8.34**	**26.24**

Table 3.3: **Open-world performance comparison based on BLEU and ROUGE scores.** OSCaR is LLaVA fine-tuned with OSCaR data, and mixed data is a combination of LLaVA data and OSCaR data.

Open World	Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
In Domain	OSCaR 13B (OSCaR data only)	5.86	28.64	8.43	24.91
	OSCaR 7B (OSCaR data only)	5.73	29.10	8.38	25.47
	OSCaR 13B (Mixed data)	**6.19**	29.36	8.74	25.69
	OSCaR 7B (Mixed data)	6.13	**30.00**	**8.95**	**26.25**
Out of Domain	OSCaR 13B (OSCaR data only)	5.32	27.20	7.62	23.67
	OSCaR 7B (OSCaR data only)	5.18	27.07	7.50	23.65
	OSCaR 13B (Mixed data)	5.24	26.18	7.36	23.09
	OSCaR 7B (Mixed data)	**5.69**	**28.99**	**8.29**	**25.38**

models have scored very low, whereas BLIP2, Video-LLaMA, and LLaVA models, which are currently the most advanced models, have achieved significant improvements. Our proposal has surpassed every previous state-of-the-art model by a large margin on these metrics.

GPT4 Evaluation: The experimental results of evaluating LLaVA, OSCaR, and GPT-4V captions on five criteria using GPT-4V are shown in Table 3.4.

According to the metric used, OSCaR performs significantly better than LLaVA. Additionally, OSCaR achieved 88.19%, 87.01%, 90.81%, 89.21%, and 97.94% in accuracy, helpfulness, detail level, reasoning, and overall, respectively, compared to GPT-4V. On av-

Table 3.4: Evaluation scores using GPT-4V under different criterion are listed in the table.

Criteria	LLaVA	OSCaR	GPT-4V
Accuracy	53.60	82.93	94.04
Helpfulness	51.63	80.78	92.83
Reasoning	53.64	79.20	87.22
Detail	40.56	87.30	89.14
Overall	51.96	80.92	90.72

erage, OSCaR is **90%** as good as GPT-4V. The visualization can be seen at Figure 3.1.

Human Study: In our study to assess caption quality from various models, seven evaluators reviewed five videos with four captions each (three for frames, one for state changes), provided by seven models. Each caption had seven different options generated by seven different models. Evaluators could select up to two options per caption that they think are the best. Figure 3.6 shows the results of this experiment. We calculated the percentage of times each model was selected and found that OSCaR achieved 73.93%, which was only 8.57% lower than GPT-4V. OSCaR significantly outperformed LLaVA by more than two times. These results demonstrate that OSCaR is a promising model for generating high-quality captions.

3.6.4 Open-world Objects Evaluation

Evaluating the performance of machine learning models solely based on objects seen during training isn't enough. To more thoroughly test their effectiveness, we also evaluated them on objects not included in the training set, representing the open world. In this part of our study, we compare the quality of text produced by our model and GPT-4V for these open-world objects, using BLEU and ROUGE scores as our metrics.

In-domain Objects Evaluation: The evaluation results on objects in the cooking domain that were not included in the instruction fine-tuning data are presented in Table 3.3. When compared with the results in Table 3.2, the overall performance is better when testing with

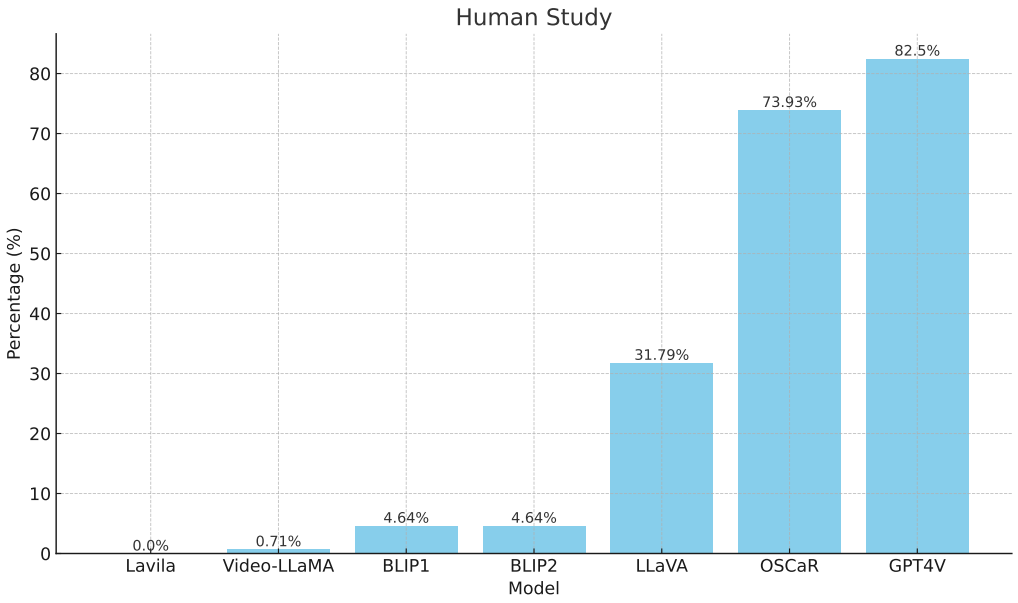


Figure 3.6: **Human study results.** The figure shows the percentage that each model was selected by participants as producing favorable descriptions in a human rating study.

in-domain open-world objects. One of the reasons for this is that the evaluation set in Table 3.2 was corrected by humans, while the data used in Table 3.3 was generated from GPT-4V. Nevertheless, the outcomes of this experiment indicate the generalizability of models when dealing with new objects.

Objects Beyond Cooking Domain: Table 3.3 presents the open-world evaluation for various domains. The dataset employed in this experiment is discussed in section 3.4.2, which comprises 356 videos from 51 distinct domains. Compared to the experiment in table 3.2, the outcomes of this experiment are generally lower. Specifically, for LLaVA 7B with mixed data, this experiment shows a decline of 0.1, 0.95, 0.05, and 0.85 on BLEU, ROUGE-1, ROUGE-2, and ROUGE-L, respectively. This decline indicates two things: 1) the open domain is challenging and may require domain-specific data for fine-tuning to achieve better performance, and 2) even in the absence of new domain data, the decrease in performance is not too significant, and showing the generalizability of our model.

Table 3.5: **Performance comparison based on BLEU and ROUGE scores in different domains.** The table compares various models with open-world benchmarks.

Domain	Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Cooking Domain	LLaVA [102]	2.56	23.77	5.64	21.08
	BLIP1 [109]	4.33×10^{-5}	0.75	0.026	0.73
	BLIP2 [97]	0.043	8.2	1	7.4
	LaViLa [105]	4.34×10^{-5}	3.09	0.27	3.07
Other Domains	LLaVA [102]	2.88	23.96	5.85	21.26
	BLIP1 [109]	7.39×10^{-5}	1.15	0.077	1.13
	BLIP2 [97]	0.028	9.04	1.05	8.2
	LaViLa [105]	6.95×10^{-5}	3.1	0.29	3.07

3.6.5 Ablation Study

Our research also examined the accuracy of video frame annotations in the EPIC-KITCHENS and Ego4D datasets. We used Amazon Mechanical Turk annotators to evaluate 500 video data points for the precision and completeness of descriptions, categorizing them into four classes. In addition, we analyzed 100 samples from each setting of zero-shot and two-shot to determine the best strategy for scaling up data annotation. Our findings indicate that zero-shot is the more effective approach for annotating our task's data.

Our findings, detailed in Table 3.6, compare zero-shot and two-shot performance in aligning descriptions with human standards of accuracy and relevance, as derived from video frame analyses. This table illustrates how well the GPT-4V model's natural language descriptions, evaluated by Amazon Mechanical Turk annotators in zero and two-shot scenarios, match human judgment. The percentages indicate the extent to which these descriptions accurately and relevantly depict the video content, based on a frame-by-frame review. Each description was judged for its thoroughness and relevance in detailing the object and its activities. Annotators followed established guidelines to determine the quality of data in their assessments.

The results reveal a notable disparity in description quality between the zero-shot and two-shot methods. The zero-shot approach yielded a higher proportion of Fully Detailed and Comprehensive descriptions, while the two-shots method indicated a greater occurrence of descriptions with errors or misleading content. This variation highlights the dif-

ferences in data quality and annotator perceptions under varying evaluation conditions, underscoring the importance of method selection in annotation studies.

Table 3.6: The table lists the distribution of Amazon Mechanical Turk annotators' choices of descriptions of objects and object state changes in 0 and two-shot tests by the GPT-4V model in %.

Satisfaction Class	Zero-shot	Two-shots
Fully Detailed	56.25	33.25
Minor Mistakes	16.75	28.25
Lacks Important Details	13.25	23.00
Hallucinating	13.75	15.50

In Table 3.5, we present the results of our experiment where we evaluate various models in open-world benchmarks, including the cooking domain and other domains. We have observed that the performance of other baselines has generally decreased in open-world benchmarks. These results demonstrate the importance of building models that can be generalized in the world. However, capturing the state of objects while dealing with diverse objects and domains is still a major challenge.

3.7 Conclusion

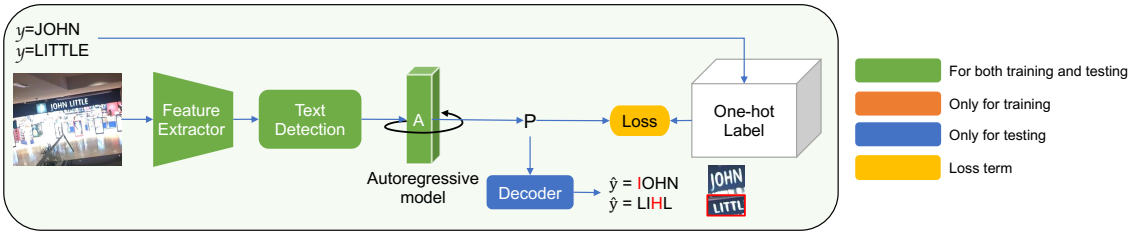
This paper presents a new task for comprehending the state of objects and their changes using natural language. We also propose a data generation pipeline that utilizes the capabilities of GPT-4V to tackle this task. Furthermore, we introduce OSCaR, a dataset that includes training data and a benchmark with various protocols. Our comprehensive experiments not only demonstrate the superiority of our methods in comparison to previous state-of-the-art open-source solutions but also examine the limitations of GPT-4V in addressing this challenge.

CHAPTER 4
EFFICIENTLY INCORPORATING LINGUISTIC PRIORS FOR SCENE TEXT SPOTTING

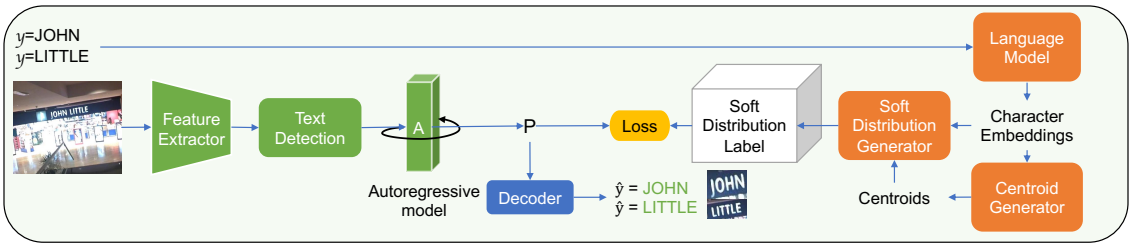
4.1 Introduction

Text detection and recognition in natural scenes is a research area of considerable importance, with diverse practical applications ranging from aiding the visually impaired [22] to facilitating robot navigation [23] and enabling mapping, and localization [24]. However, many text instances in natural settings exhibit inherent ambiguity stemming from aesthetic variations, environmental deterioration, or poor illumination conditions. This ambiguity can often be partly reduced by considering a list of lexicons or a dictionary [25]. However, this usually increases the complexity of models.

An end-to-end text spotting system typically involves two main steps: text detection and text recognition. Over the years, researchers have devoted considerable effort to advancing the state of the arts in both text detection [26, 27, 28, 29, 30, 31, 32] and text recognition [33, 34, 35, 36, 37, 38]. Despite advancements, current scene text spotting models do not match human reading capabilities, especially in recognizing distorted or blurry characters. This remarkable feat is due to the fact that linguistic knowledge provides a powerful prior that can help disambiguate text. Existing autoregressive text recognition models [39, 40, 41, 42] can leverage some of the linguistic structure from the training data in the decoder component, where the model predicts the next character by using previous character predictions as inputs. However, these approaches have yet to fully use linguistic knowledge for the following reasons. Firstly, existing scene text datasets are limited, making effective linguistic knowledge capture challenging. Secondly, current models employ basic one-hot vector encoding, ignoring the uncertainty or relationship between charac-



(a) The normal scene text spotting pipeline



(b) The **proposed** scene text spotting pipeline

Figure 4.1: **Traditional spotting pipeline (a) and proposed pipeline (b) on training.** In the traditional pipeline, models use the one-hot label directly to guide the training for the scene text system. Our proposal replaces the one-hot encoding by using soft distributions for every label character and improving detection and recognition results. Besides, we proposed a method to leverage knowledge from pretrained language models and construct the soft distribution well-adapted to the scene text domain without finetuning language models.

ters. Furthermore, autoregressive models predict character distribution based on previous characters, contrasting traditional methods that align this with one-hot, which has no connections to other characters. Thus, better modeling of linguistic knowledge is needed in pipelines that rely solely on autoregressive models.

During inference, text detection models often fail if the distance between two characters of the same word is too far (detecting them as two different words), or if the distance between two different words is too close (detecting them as one word), or if the word length is too long (missing some characters). Traditional methods use a one-hot label, which can encourage better learning in detection when considering recognition as a constraint for detection models. However, this method only looks at each character independently, without considering the context of the entire word. This limits the ability of the detection model to look at the entire word in a comprehensive way. Linguistic priors allow detection models to look at the entire word, which helps to improve detection performance. For

example, we know that words are typically made up of multiple characters, and that the characters in a word are typically spaced close together. By taking linguistic priors into account, detection models can better understand the context of the entire word and make more accurate predictions.

In this paper, we propose a novel end-to-end scene text spotting method that efficiently incorporates language priors from trained language models. Our approach can also be viewed as a multimodal knowledge distillation technique, which has yet to be fully utilized for scene text spotting. By directly leveraging language model output as label embeddings to guide text spotting learning, our approach provides more detailed guidance than traditional one-hot encoding. We empirically tested our approach on multiple benchmark datasets, including Total-Text [110], SCUT-CTW1500 [111], and ICDAR2015 [112], demonstrating its remarkable effectiveness in transferring knowledge from language models to scene text spotting models. Besides, to demonstrate the generalizability of our proposal, we also implemented our method on a state-of-the-art scene text recognition pipeline. Notably, our method allows for direct utilization of language model knowledge by scene text models without requiring any fine-tuning in the specific domain of scene text data.

Our contributions are threefold:

- We demonstrate the efficacy of leveraging language knowledge derived from a large language model to enhance the performance of both scene text spotting and recognition models. Specifically, we illustrate how the incorporation of language knowledge through the technique of knowledge distillation can benefit both text detection and text recognition tasks.

- We present a novel and intuitive approach for integrating language models into the scene text spotting and recognition learning process without increasing model complexity in training and testing. Our proposed approach has been shown to be highly effective and is supported by empirical validation.

- We introduce an innovative method that enables the utilization of pre-trained language model representations without the need for domain-specific fine-tuning on the scene text data, thus increasing the applicability and efficiency of the scene text spotting and recognition model.

4.2 Related Works

Scene Text Auto-regression Model. Auto-regressive models are widely used to tackle scene text recognition [54, 39, 53, 55, 40, 41, 42]. For example, CRNN [53] applied a deep bidirectional LSTM to infer the output probability distribution of the text. A CTC loss function removes duplicate characters in the output text series. In addition to CRNN, SCATTER [56] sets up an attention layer to capture the intra-sequence relationships. Charnet [57] fuses character-level and word-level encodings over the input to make the output more robust. Taking good practice in past research, we adopt an auto-regressive method that decodes output sequences at the character level. A canonical RNN can execute the auto-regressive process with a unique direction from left to right.

Language-driven for Scene Text Recognition. The integration of language knowledge into scene text recognition models has emerged as a popular research direction [34, 61, 62, 63]. For example, VisionLAN [63] proposed a visual reasoning module that simultaneously captures visual and linguistic information by masking the input image at the feature level. Similarly, ABINet [61] utilized a language model for iterative processing to enhance recognition accuracy with each iteration. Additionally, LevOCR [62] learned a refinement policy with insertion and deletion actions to iteratively enhance recognition accuracy. However, these approaches face two common challenges. Firstly, they increase the number of parameters in the models, which can lead to computational inefficiencies or slower model performance when iterative running is required. Secondly, these methods often rely on limited exploitation of language knowledge from small-scale scene text data without fully leveraging the vast knowledge from large-scale language models. ABI-

Net++, an extension of ABINet, incorporates linguistic knowledge into scene text spotting models. However, like ABINet, this approach adds complexity through iterative execution, thereby elevating the model's complexity. The utilization of language knowledge for scene text spotting remains relatively limited. To the best of our knowledge, our work represents the first attempt to leverage language knowledge from large language models without increasing model complexity or additional computational requirements during training and inference. This efficient approach can improve both text spotting and detection.

Knowledge Distillation. Using knowledge transfer from a teacher to a student model can enhance performance at low cost [64]. Typically, the teacher's output probabilities guide the student model through a loss function optimization. DeiT [113] distinguishes between soft and hard-label distillation. Soft distillation directly uses the teacher's predictions, while hard-label distillation converts these to one-hot encoding. In this work, we propose a cross-modal knowledge distillation to transfer knowledge from text to image for scene text spotting.

4.3 Language-guided Scene Text Spotting

We introduce a novel approach to enhance in-the-wild scene text spotting by integrating prior language knowledge. Scene text spotting locates text instances and computes their feature maps from an image. Using an auto-regressive model, we produce a probability map for each text instance, representing character distributions. We then compute embeddings for text labels and establish character prototypes. Using these embeddings and character prototypes, we create a soft distribution for each character. Ultimately, we obtain a distribution matrix for each word label, where each column represents a soft distribution corresponding to one character in the word. Subsequently, we employ these distributions as soft label distributions for the training process.

In this section, we describe the pipeline for scene text spotting, highlighting how the soft label distribution is created. Additionally, we discuss our loss function. Section

Autoregressive-based Scene Text Recognition presents an in-depth description of the auto-regressive model for scene text recognition. Furthermore, in section Character Embedding, we delve into the topic of character embedding. Our approach for centroids estimation and soft distribution generation will be introduced below.

4.3.1 Autoregressive-based Scene Text Recognition

Auto-regressive models generate outputs over time, using past predictions as current inputs, described as $P(x_t|x_{t-1}, ..., x_1)$. Traditional training uses one-hot labels to represent characters, where a character c might be shown as $[0, 0, 1, ..., 0]$. However, this method fails to capture relationships between characters in natural language contexts. For example, it does not represent the likelihood of character x following b being less than o after b. Although widespread in natural language, this knowledge cannot be effectively represented using one-hot encoding. In other words, during training, the model approximates a conditional distribution with a probability distribution to represent the character, irrespective of the context of the characters surrounding it.

The auto-regressive model used in training captures limited contextual knowledge due to scarce scene text datasets. Despite pre-training with synthetic datasets, the information learned is constrained by the dictionary used for scene text image generation. Moreover, during the fine-tuning process, the model tends to forget the limited knowledge captured from the pre-training data. Additionally, the recognition step of the scene text spotting pipeline is often implemented with lightweight models, restricting their ability to generalize with language. To address these challenges, we propose augmenting the scene text spotting model with a language model, which can extract knowledge from vast amounts of text data and enhance the model's ability to learn image patterns and language in a more comprehensive way.

4.3.2 Character Embedding

Language models require converting text sequences into numeric embeddings through tokenization, either at the word or sub-word level. For tasks like scene text spotting, which prioritize learning character distributions, a character-level pre-trained model is more effective. CANINE [114], a recent model, differs from traditional tokenizers such as Byte Pair Encoding, WordPiece, and SentencePiece, as it employs a Transformer encoder trained directly on Unicode characters. Although this increases sequence length, CANINE addresses this issue with an efficient downsampling technique before applying the deep Transformer encoder. The CANINE model consists of a downsampling function, ENCODE, and an upsampling function, which operate on an input sequence of character embeddings e $\in R^{n \times d}$ with length n and dimension d, resulting in the output sequence embedding $\mathbf{Y}_{seq}$:

$$\mathbf{Y_{seq}} \leftarrow \text{Up(ENCODE(DOWN(e))).} \tag{4.1}$$

To ensure that different characters do not have the same representation, the model uses a generalized hashing approach that concatenates the representations associated with various hash values. In our work, we utilize the pre-trained embedding $\mathbf{Y}_{seq}$ to estimate the centroid for each character, which is then considered the prototype for the embedding clusters.

4.3.3 Centroid Generation

To generate a probability distribution from a given representation, a matrix multiplication operation is performed between the representation vector, denoted by **x**, and a weight matrix, denoted by **W**, which produces a vector of logit values. The logit vector is then fed through a softmax function, resulting in a probability distribution vector, denoted by **D**, which represents the probability of the representation belonging to each character in the vocabulary. The weight matrix **W** is a prototype matrix, where each column corresponds to a prototype of a character in the vocabulary. The similarity between a representation and

each character prototype in the latent space needs to be calculated in order to determine the most appropriate label for the representation. This similarity measurement allows for the selection of the label that is most compatible with the embedding.

$$D_i = \frac{\exp(W_i^T x_j)}{\sum_{i=1}^{k} \exp(W_i^T x_j)}, \tag{4.2}$$

where W_i is i^{th} column of prototype matrix, x_j is the representation of the character j^{th} in the word, and k is the size of character vocabulary.

In our approach, selecting an optimal **W** in Equation (4.2) is crucial, where columns serve as character prototypes. Using the W-matrix directly from the language model might yield a too general probability distribution, causing disparity between scene text data and text data utilized to train the language model. Additionally, fine-tuning the language model with scene text data can introduce biases, compromising the model's generality. Furthermore, the process of fine-tuning the language model is time-consuming. In this paper, we propose using each prototype as the centroid of its character representation cluster by averaging all character representations of the same cluster:

$$W_i = \frac{1}{n} \sum_{j=1}^{n} x_{ij}, \tag{4.3}$$

where x_{ij} is the j^{th} representation of character i^{th}, which is equivalent to W_i in prototype matrix.

4.3.4 Soft Distribution Generation

Using the matrix **W** from equation 4.3, we derive the soft label distribution for training from equation 4.2. Our method bridges the gap between the pretrained language model and the scene text dataset without the need for fine-tuning. The texts we use to generate the W matrix have the trade-off between scene text dataset and text from outside dataset. Basically, this is a sampling process to approximate the real centroids of language model.

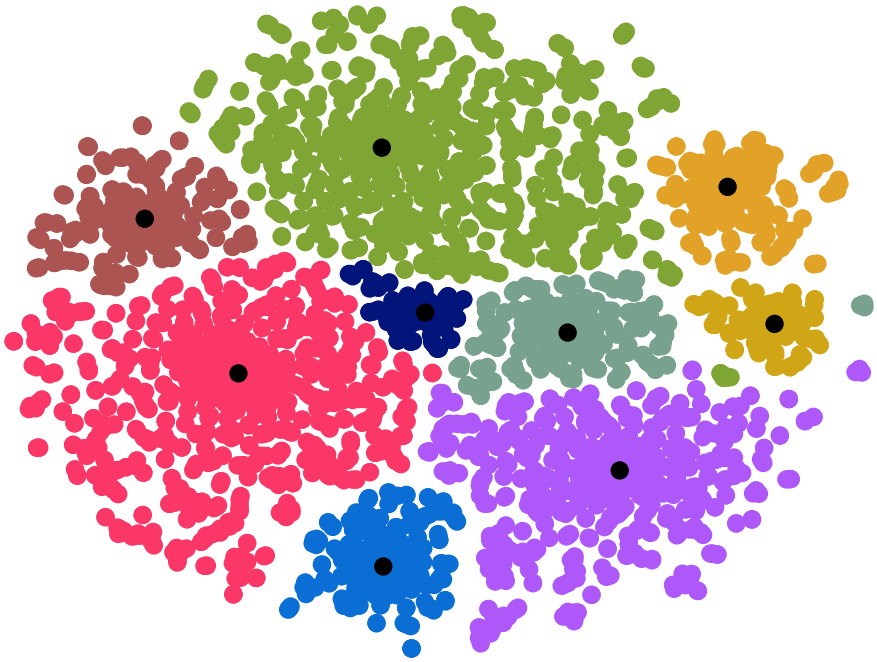


Figure 4.2: **Centroid Estimation.** Visualization of character embedding for 9 characters $a, b, c, d, e, f, g, h, i$. Each cluster equivalent with a character, and black points in the center are the centroids generated by (4.3).

Using only the scene text dataset risks overfitting and losing general language knowledge. Conversely, using many external words keeps the centroids close to the original language model weights. In this work, we generate the centroids by a subset of words from a generic dictionary [115], and the whole word appeared in the scene text training dataset. We will discuss more about our settings in the experiments part. The predicted distribution of the model will be denoted by distribution **P**. We use KL-Divergence as the loss function to measure the difference between the predicted distribution and our soft label distribution:

$$KL(D||P) \propto -\sum_{i=1}^{k} D_i \log(P_i). \tag{4.4}$$

Finally, we sum the loss of all columns together as our final loss function:

$$\mathcal{L}(x, y) = \sum_{i=1}^{l} KL(D^{(i)}||P^{(i)}). \tag{4.5}$$

where $D^{(i)}$ and $P^{(i)}$ are i^{th} predicted distribution and soft label distribution, respectively. l is the longest length over all words.

During the creation of the soft distribution label, we perform post-processing to remove noise. First, we check whether the distributions can represent mislabels by checking which character the highest probability belongs to. Next, distributions are filtered using a threshold T; those with a label probability greater than or equal to T are retained. For those that don't meet the threshold, a predefined distribution is used. In positions with a one-hot representation of 1, we set the value to T, and all other positions are set to P_i:

$$P_i = \frac{1-T}{k-1}, \tag{4.6}$$

where T is the probability threshold we set for the label, P_i is the probability value of the characters that are not label, and k is the size of the character vocabulary.

The parameter T greatly influences generated distributions. A low T can yield overly flat distributions with high entropy, while a high T may force shifts to non-linguistic-based ones. We determine T using text labels. A 0.85 threshold effectively curbs excessive entropy while retaining language uncertainty. With this threshold, the error rate across datasets was satisfactory ($\leq$0.8%), and any errors were post-processed. This post-processing is done once before training, removing the need for recalculations later. Our approach is straightforward, with the potential for future improvements in threshold selection.

4.3.5 Implementation Details

In our scene text spotting approach, we used the CANINE [114] model to extract character representations from the dictionary. We then averaged embeddings of the same class from a vocabulary combining the dataset and 30k random words from a general dictionary [115] to produce centroids. Details about the dictionary used in scene text recognition are shown in Scene text recognition experiment section. Soft distributions were derived from these

embeddings and centroids, which we later refined. A threshold T in Eq.(4.6) of 0.85 was applied to modify soft distribution probabilities. In Mask TextSpotterv3, we only applied our method to its spatial attention module, an auto-regressive-based model, for consistent comparison. Adam optimizer [116] was used for optimization.

4.4 Experiments

This section describes our method effectiveness on both scene text spotting (TotalText [110], SCUT-CTW1500 [111], and ICDAR2015 [112]) and scene text recognition (IIIT5k [117], SVT [118], ICDAR2013 [119], SVTP [120], ICDAR2015 [112], CUTE80 [121]) datasets. Besides, we will also describe experiment settings and datasets used for both pretraining and finetuning. Figure 4.3 shows how our proposed method has improved baselines.

4.4.1 Scene Text Spotting Experiments

In our approach, modifying the underlying model architecture is unnecessary, making it non-mandatory to re-train the model with synthetic datasets. This implies that the existing pretrained model available in the baseline repositories can be employed. This aspect is especially important because re-training using extensive synthetic datasets often necessitates substantial computational resources and extended training durations. Empirical evidence suggests that our methodology substantially enhances the baseline by directly fine-tuning on target datasets, eliminating the need for re-train with the synthetic dataset. Furthermore, to investigate the feasibility of integrating linguistic insights during the pretraining phase, we re-trained ABCNetV2 using soft linguistic labels on Curved Synthetic, TotalText [110], ICDAR 15 [112], ICDAR 13 [119], MLT-2017 [122], and TextOCR [123] datasets. We have termed this variant External Dataset (ED) and reported its outcomes with the results procured from direct fine-tuning on the provided checkpoints.

In our scene text spotting experiments, first, we removed all words present in the test set from a generic 90k dictionary. Next, we sampled a subset of words from this newly formed

(a) ABCNetv2: AREAT
ABCNetv2+L: GREAT

(b) ABCNetv2: ENISIAD
ABCNetv2+L: SPANISH

(c) ABCNetv2: KARAOKS
ABCNetv2+L: KARAOKE

(d) ABCNetv2: HONCKONG
ABCNetv2+L: HONGKONG

Figure 4.3: **Qualitative results on Total-Text dataset.** Our approach is more capable of recognizing scene texts than the baseline. These outputs are directly taken from the model when the dictionary is not used in the testing phase.

dictionary and merged it with the collection of words found in the training set to generate centroids. This step was implemented to ensure that the model would not unintentionally be exposed to words exclusive to the test set that had not appeared in the training set. For each dataset, we train a model using all the words from its training set. Subsequently, we use a single checkpoint to evaluate the model across all dictionaries associated with that dataset.

Experiment on Total-Text

The Total-Text dataset [110], introduced in 2017, is a prominent benchmark for arbitrarily shaped scene text. Consisting of 1,555 photos (1,255 for training and 300 for testing), it features images with low-contrast backgrounds and complex texts. The dataset predomi-

Method	None	Full
Charnet [124]	66.6	-
SwinTextSpotter [125]	74.3	84.1
PAN++ [126]	68.6	78.6
GLASS [127]	76.6	83.0
TTS [128]	75.6	84.4
CRAFTS [129]	**78.7**$^{(*)}$	-
SPTS [130]	74.2	82.4
TESTR [131]	73.3	83.9
ABINet++ [65]	77.6	**84.5**
Mask TextSpotterv3 [132]	71.2	78.4
Mask TextSpotterv3+L **(Ours)**	**73.4**	**81.2**
ABCNetv2 (github checkpoint)	71.8	83.4
ABCNetv2+L **(Ours)**	74.5	84.6
ABCNetv2+L (ED) **(Ours)**	**76.8**	**87.1**$^{(*)}$

Table 4.1: **Scene text spotting results on Total-Text**. The values shown in the table are H-mean scores for end-to-end models. *None* and *Full* represent without and with a dictionary, respectively; the dictionary contains all testing words in the inference phase. ED denotes for re-train with external data. Our methods significantly improved upon the baselines, ABCNetv2 and Mask TextSpotterv3, surpass ABINet++ when using a full dictionary with ABCNetv2+L (directly fine-tuning from provided checkpoint) and this improvement is even more significant when re-train with external data (ED). **(*)** denotes the best score. We report scores wherever they are available on paper or GitHub.

nantly includes irregular text, with each photo containing at least one curved word. Text instances are annotated with polygons, and its extended version further annotates each instance with ten fixed points and a recognition sequence. The dataset exclusively contains English text.

Table 4.1 shows the performance of our ABCNetv2+L and Mask TextSpotterv3+L methods in both terms of using and not using the full dictionary during the inference process. The full dictionary includes all words that appeared in the test set. The ABCNetv2 [133] results in Table 4.1 are the results reported by the authors in the paper. ABCNetv2 [133] (GitHub checkpoint) are the results of the checkpoint published by the authors on their GitHub. We use this implementation as the baseline for our proposal. Checkpoint results generally produce better results than the performance reported in the paper. We

compare our results with reported and checkpoint-generated results to make a fair comparison. It can be seen that our proposal improved both ABCNetv2 and Mask TextSpotterv3 significantly in both scenarios with and without full dictionary. Moreover, ABCNetv2+L with dictionary and finetuned on provided checkpoint outperformed current state-of-the-art ABINet++ [65] by 0.1. This improvement has been increased to 2.6 when doing re-train with ED using our method.

Experiment on ICDAR 15

Images from the ICDAR 2015 [112] dataset were unintentionally taken from the real world through Google Glass. In contrast to earlier ICDAR 13 datasets [119], when the text was neat, well-captured, and horizontally centered in the photos. The dataset consists of 500 testing photos with complex backgrounds and 1000 training images. Additionally, some text may be small or low resolution, appearing anywhere and in any orientation. The annotation only contains English samples and is based on word level.

Images from the ICDAR 15 [112] dataset were unintentionally taken from the real world through Google Glass. In contrast to earlier ICDAR 13 datasets [119], which featured cleanly captured text horizontally centered in the images, the ICDAR 15 dataset comprises 1000 training images and 500 testing photos that present significant challenges to text recognition algorithms. The images in the dataset feature complex backgrounds and text may appear in any orientation and at varying resolutions. Furthermore, the dataset is exclusively comprised of English text samples annotated at the word level.

Table 4.2 shows the experimental results of our proposal compared with previous works. For ICDAR 15, we use three scenarios to measure and evaluate the model, including strong, weak, and generic dictionary. Correspondingly, the strong dictionary for each image is a set of 100 words taken from the test set, in which all the words appearing in the image are in this group of 100 words. The weak dictionary is all the words that appear in the test set, and the generic vocabulary is the set of words that can appear, including a dictionary

Table 4.2: **Scene text spotting results on ICDAR 15**. The values shown in the table are H-mean scores for end-to-end models. S, W, and G represent Strong, Weak, and Generic dictionaries used in the inference phase, respectively. Our method improved the baselines in all settings. Incorporating our method into ABCNetv2+L with ED, we outperformed current state-of-the-art on both Strong and Weak dictionary settings. **(*)** denotes the best score.

Method	S	W	G
SwinTextSpotter [125]	83.9	77.3	70.5
PAN++ [126]	82.7	78.2	69.2
GLASS [127]	84.7	80.1	76.3
MANGO [134]	85.4	80.1	73.9
CRAFTS [129]	83.1	82.1	74.9
TTS [128]	85.2	81.7	77.4
SPTS [130]	77.5	70.2	65.8
TESTR [131]	85.2	79.4	73.6
ABINet++ [65]	86.1	81.9	77.8
DeepSolo [135]	**88.1**	**83.9**	**79.5**$^{(*)}$
Mask Textspotterv3 [132]	83.3	78.1	74.2
Mask TextSpotterv3+D [25]	85.2	81.9	75.9
Mask TextSpotterv3+L **(Ours)**	**85.9**	**81.9**	**77.4**
ABCNetv2 (github checkpoint)	83.7	78.8	73.2
ABCNetv2+L **(Ours)**	85.1	80.9	75.4
ABCNetv2+L(ED) **(Ours)**	**88.4**$^{(*)}$	**84.4**$^{(*)}$	**78.6**

of 90k words [115]. Our proposal effectively enhances the performance of the baseline models across all three types of dictionaries without any additional complexity. Notably, our ABCNetv2+L(ED) surpassed DeepSolo on strong and weak dictionaries by 0.3 and 0.5, and achieved state-of-the-art results on these dictionaries. Furthermore, while directly applying of our method to the pretrained model showed improvement, integrating it with external datasets (ED) in line with recent state-of-the-art elevated its performance remarkably.

Experiment on SCUT-CTW1500

The SCUT-CTW1500 [111] dataset, popular for arbitrary-shaped scene text, differs from Total-Text by including both Chinese and English. Annotations are at the text-line level, accommodating documents with stacked small text segments. It has 1000 training and

Table 4.3: **Comparison of scene text recognition accuracy on six datasets.** TargetDict denotes the list of words present in training sets of IIIT5k, SVT, IC13, IC15, SVTP, and CUTE datasets. The top-2 results are highlighted.

Method	Training Data	Regular Text			Irregular Text		
		IIIT	SVT	IC13	SVTP	IC15	CUTE
ASTER [136]	MJ+ST	93.4	89.5	91.8	78.5	76.1	79.5
DAN [137]	MJ+ST	94.3	89.2	93.9	80.0	74.5	84.4
RobustScanner [138]	MJ+ST	95.3	88.1	94.8	79.5	77.1	90.3
SAR [139]	ST+MJ	91.5	84.5	91.0	76.4	69.2	83.3
SEED [140]	ST+MJ	93.8	89.6	92.8	81.4	80.0	83.6
ABINet [61]	MJ+ST+Wiki	96.2	93.5	**97.4**	89.3	86.0	89.2
S-GTR [141]	ST+MJ	95.8	94.1	96.8	87.9	84.6	92.3
LevOCR [62]	ST+MJ	96.6	92.9	96.9	88.1	86.4	91.7
SIGAT [142]	ST+MJ	96.6	<u>95.1</u>	96.8	90.5	83.0	<u>93.1</u>
MGP-STR [35]	ST+MJ	96.4	94.7	<u>97.3</u>	91.0	87.2	90.3
PARSeq [34]	ST+MJ	**97.0**	93.6	96.2	82.9	**88.9**	92.2
CornerTransformer [33]	ST+MJ	95.9	94.6	96.4	91.5	86.3	92.0
CornerTransformer+L(**Ours**)	ST+MJ	96.4	95.0	96.9	<u>91.8</u>	87.0	92.7
CornerTransformer+L(**Ours**)	ST+MJ+TargetDict	<u>96.7</u>	**95.4**	**97.4**	**92.2**	<u>87.8</u>	**93.4**

500 test images. For the SCUT-CTW1500 dataset, labels include word-level annotations and groups of text instances. Language knowledge is vital for accurate results, capturing both intra-word and inter-word relationships in long sequences. Table 4.4 shows our ABCNetv2+L notably outperforms ABCNet2, especially with a dictionary during inference. Our method with ED surpasses all previous state-of-the-art models for SCUT-CTW1500 using a strong dictionary, highlighting the effectiveness and versatility of the proposed approach in addressing the challenges associated with localizing and recognizing scene text in the long line with arbitrary shapes.

Detection Results

Table 4.5 presents quantitative evidence of the effectiveness of our proposed method in enhancing not only text recognition but also text detection. In addition, Figure 4.4 provides a visual representation of the qualitative improvements in text detection achieved through

Table 4.4: **Scene text spotting results on SCUT-CTW1500**. The values shown in the table are H-mean scores for end-to-end models. None and Strong represent without and with a strong dictionary in the inference phase, respectively. Our method improved the baselines in all settings and achieved state-of-the-art when evaluating without a dictionary for post-processing. **(*)** denotes the best score.

Method	None	Strong
TextDragon [143]	39.7	72.4
MANGO [134]	58.9	78.7
SwinTexSpotter [125]	51.8	77.0
Text Perceptron [144]	57.0	-
ABINet++ [65]	60.2	80.3
TESTR [131]	56.0	**81.5**
DeepSolo [135]	**64.2**$^{(*)}$	81.4
ABCNetv2 [133]	57.5	77.2
ABCNetv2+L **(Ours)**	59.1	78.4
ABCNetv2+L(ED) **(Ours)**	**61.2**	**81.8**$^{(*)}$

Table 4.5: Detection H-mean score comparison between ABCNetv2 and ABCNetv2+L. Our method improves detection performance on all three datasets.

	TotalText	ICDAR15	CTW1500
ABCNetv2	87.2	88.2	85.0
ABCNetv2+L (**Ours**)	88.5	88.6	85.4
ABCNetv2+L (ED) (**Ours**)	**88.9**	**89.1**	**85.8**

our approach. It is worth noting that unlike conventional object detection tasks, text detection is particularly sensitive since even the slightest inaccuracies in localizing words or including extraneous details can lead to erroneous recognition outcomes. To address this issue, we leverage language knowledge as a guiding signal to enhance the accuracy of our text detection model. Our experimental results, both quantitative and qualitative, clearly demonstrate the effectiveness of this approach in improving the overall quality of our text detection model. In summary, our simple yet effective method can significantly enhance text detection accuracy, which, in turn, directly influences text recognition outcomes.

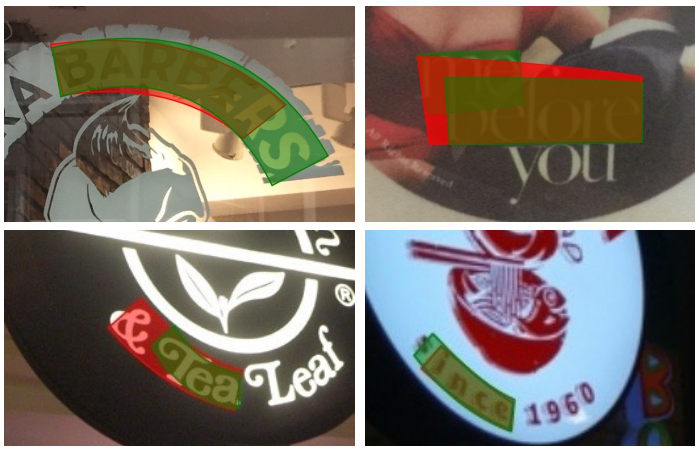


Figure 4.4: Comparison of detection results **with** (green shaded) and **without** (red shaded) language knowledge prior guidance. Language prior is not only helpful for text recognition but also for text detection.

4.4.2 Scene Text Recognition Experiments

Our approach integrates linguistic knowledge into scene text models, enabling comprehensive generalization for text recognition. Using the CornerTransformer [33] as a baseline, we train our model on SynthText and MJSynth datasets, following previous state-of-the-art methods. While image acquisition and labeling incur high costs, collecting domain-specific words is simpler. Our method capitalizes on this by generating centroids from in-domain text. We train the model with two centroid types: one using a 90k-word dictionary for fair comparison, and another incorporating words from training sets (excluding test-only words) of IIIT5k, SVT, IC13, IC15, SVTP, and CUTE. We call this list of in-domain words is target dictionary (TargetDict).

The experimental results shown in Table 4.3 highlight the efficacy of our proposed method in enhancing the accuracy of the baseline model. This improvement is consistently observed across all six benchmark datasets when trained on MJSynth and SynthText datasets, following previous approaches. Compared with a method also incorporating linguistic knowledge into scene text recognition, ABINet, our method helps CornerTransformer surpass ABINet on IIIT5k and increases the existing improvement on SVT, SVTP, ICDAR 15, and CUTE datasets more significantly. This superior performance is achieved

without adding complexity to the model in training or inference.

Additionally, generating centroids with words from TargetDict increased the improvement by a large margin. Specially, CornerTransformer+L has improved the baseline's accuracy on IIT, SVT, ICDAR 13, SVTP, ICDAR 15, and CUTE by 0.8%, 0.8%, 1%, 0.7%, 1.5%, and 1.4%. Our method with TargetDict achieves state-of-the-art on SVT, ICDAR 13, SVTP, and CUTE, and secures the second-best results on ICDAR 15 and IIIT5k. These results underscore the model's effectiveness in bridging the domain gap between synthetic and real data. Importantly, our approach leverages target domain knowledge only from the text in the target domain, minimizing the need for resource-intensive image collection and annotation processes.

4.5 Conclusion

We proposed a novel approach to transfer language knowledge to autoregressive-based scene text spotting models. Our approach overcomes ambiguous words using soft distributions based on character representation. We also proposed a process to generate soft distributions suitable for dictionaries without finetuning. Our approach was implemented on two scene text spotting backbones: ABCNetv2 and Mask TextSpotterv3, and one scene text recognition backbone: CornerTransformer, showing its generality. Extensive experiments on scene text spotting and recognition datasets can validate the effectiveness of our approach.

CHAPTER 5
DISCUSSION AND FUTURE WORK

Our investigation into the state understanding of objects and scene text recognition has yielded significant insights while also illuminating areas that require further development. This section outlines our key findings, discusses existing limitations, and proposes several directions for future research efforts.

Lack of audio integration: One notable gap in our current research is the absence of audio data integration. Audio signals often contain crucial information about the status or changes in objects, such as operational noises from machinery or alerts that could indicate malfunctions or operational statuses. Future work should focus on developing multimodal models that can effectively integrate audio and visual inputs. This holistic approach could significantly enhance the accuracy and robustness of object state analysis, particularly in complex environments like industrial settings or during interactive tasks in augmented reality systems.

Challenges in long-term state transition tracking: Our existing models, primarily based on large language models, struggle with accurately tracking and predicting long-term changes in object states. This is a critical limitation as objects in real-world scenarios often exhibit changes over extended periods. Future research should aim to develop or adapt existing models that are capable of understanding and predicting these long-term state transitions. Approaches could include enhancing models, utilizing state-of-the-art sequential processing techniques, or exploring new forms of temporal neural architectures designed to handle prolonged data sequences.

Reliance on GPT-4V's imperfect outputs: Although GPT-4V has shown strength in generating data for this research problem, its outputs are imperfect. This limitation highlights the need for strategies to efficiently learn from and improve upon the imperfect data pro-

vided by GPT-4V.

Advanced Scene Text Spotting Techniques: The recognition of textual content on objects plays a pivotal role in our research, given its potential to reveal significant information about the object's state or context. Improving scene text recognition, or OCR technology, remains a vital area for future research. This involves not only enhancing the accuracy of text detection and recognition in varied and challenging conditions but also integrating these capabilities with object state analysis. Such advancements could lead to better performance in environments with poor lighting, atypical angles, or partial occlusions.

Dynamic Analysis of Object Interactions: Thus far, our research has primarily addressed static object states. However, understanding the dynamic interactions between objects, particularly in motion, could provide deeper insights into their behavior and changes. Future research could explore how objects' states are influenced by their interactions and movements within an environment. This direction is particularly relevant for developing smarter systems in autonomous navigation, robotics, and interactive simulations, where understanding the interplay between multiple moving objects is crucial.

Addressing Real-world Application Challenges: Finally, applying these research findings in real-world scenarios presents its own set of challenges, including the scalability of solutions, their adaptability to different environments, and the handling of real-time data. Future work should also consider these practical implementation aspects, focusing on creating solutions that are not only theoretically sound but also practically viable and robust under diverse conditions.

By exploring these areas, future research can significantly expand the scope and impact of object-understanding technologies, making them more effective and applicable across a wider range of real-world applications.

REFERENCES

[1] D. Damen *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," *ArXiv*, vol. abs/1804.02748, 2018.

[2] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1346–1353, 2012.

[3] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2847–2854, 2012.

[4] G. A. Sigurdsson, A. K. Gupta, C. Schmid, A. Farhadi, and A. Karteek, "Charades-ego: A large-scale dataset of paired third and first person videos," *ArXiv*, vol. abs/1804.09626, 2018.

[5] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *European Conference on Computer Vision*, 2018.

[6] K. Grauman *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18 973–18 990, 2021.

[7] K. Grauman *et al.*, "Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives," *ArXiv*, vol. abs/2311.18259, 2023.

[8] B. Cai, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Understanding human-object interactions in rgb-d videos for human robot interaction," *CVPR*, 2016.

[9] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. Mayol-Cuevas, "You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video," in *BMVC*, 2016.

[10] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition," in *ICCV*, 2019.

[11] X. Zhou and D. Ramanan, "Temporal localization of fine-grained actions in videos by domain transfer from web images," *ACM Multimedia*, 2015.

[12] K. K. Singh, K. Fatahalian, and A. A. Efros, "Krishnacam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks," *IEEE Winter Conference on Applications of Computer Vision*, 2016.

[13] A. Furnari, S. Battiato, and G. M. Farinella, "Rolling-unrolling lstms for action anticipation from first-person video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2272–2285, 2020.

[14] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1346–1353.

[15] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2714–2721, 2013.

[16] H. Jiang and K. Grauman, "Seeing invisible poses: Estimating 3d body pose from egocentric video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[17] E. Ng, D. Xiang, H. Joo, and K. Grauman, "You2me: Inferring body pose in egocentric video via first and second person interactions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[18] T. Souček, J.-B. Alayrac, A. Miech, I. Laptev, and J. Sivic, "Look for the change: Learning object states and state-modifying actions from untrimmed web videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 956–13 966.

[19] J.-B. Alayrac, I. Laptev, J. Sivic, and S. Lacoste-Julien, "Joint discovery of object states and manipulation actions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2127–2136.

[20] Z. Xue, K. Ashutosh, and K. Grauman, "Learning object state changes in videos: An open-world perspective," *ArXiv*, vol. abs/2312.11782, 2023.

[21] Y. Niu, W. Guo, L. Chen, X. Lin, and S. fu Chang, "Schema: State changes matter for procedure planning in instructional videos," *ArXiv*, vol. abs/2403.01599, 2024.

[22] H. Jabnoun, F. Benzarti, and H. Amiri, "A new method for text detection and recognition in indoor scene for assisting blind people," in *International Conference on Machine Vision*, 2017.

[23] S. Liu, H. Xu, Q. Li, F. Zhang, and K. Hou, "A robot object recognition method based on scene text reading in home environments," *Sensors (Basel, Switzerland)*, vol. 21, 2021.

[24] J. Greenhalgh and M. Mirmehdi, "Recognizing text-based traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 1360–1369, 2015.

[25] N. L. Nguyen *et al.*, "Dictionary-guided scene text recognition," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7379–7388, 2021.

[26] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9357–9366, 2019.

[27] J. Tang, W. Qian, L. Song, X. Dong, L. Li, and X. Bai, "Optimal boxes: Boosting end-to-end scene text recognition by adjusting annotated bounding boxes via reinforcement learning," in *European Conference on Computer Vision*, Springer, 2022, pp. 233–248.

[28] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 20–36.

[29] S.-X. Zhang *et al.*, "Deep relational reasoning graph network for arbitrary shape text detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9699–9708.

[30] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3123–3131.

[31] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 11 474–11 481.

[32] J. Liu, X. Liu, J. Sheng, D. Liang, X. Li, and Q. Liu, "Pyramid mask text detector," *arXiv preprint arXiv:1903.11800*, 2019.

[33] X. Xie, L. Fu, Z. Zhang, Z. Wang, and X. Bai, "Toward understanding wordart: Corner-guided transformer for scene text recognition," in *European Conference on Computer Vision*, 2022.

[34] D. Bautista and R. Atienza, "Scene text recognition with permuted autoregressive sequence models," *ArXiv*, vol. abs/2207.06966, 2022.

[35] P. Wang, C. Da, and C. Yao, "Multi-granularity prediction for scene text recognition," in *European Conference on Computer Vision*, 2022.

[36] L. Zhao, Z. Wu, X. Wu, G. Wilsbacher, and S. Wang, "Background-insensitive scene text recognition with text semantic segmentation," in *European Conference on Computer Vision*, 2022.

[37] C. Liu, C. Yang, and X.-C. Yin, "Open-set text recognition via character-context decoupling," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4513–4522, 2022.

[38] A. K. Bhunia, P. N. Chowdhury, A. Sain, and Y.-Z. Song, "Towards the unseen: Iterative text recognition by distilling from errors," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14 930–14 939, 2021.

[39] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4168–4176.

[40] J. Wang and X. Hu, "Gated recurrent convolution neural network for ocr," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[41] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "Aon: Towards arbitrarily-oriented text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5571–5579.

[42] C. K. Chng *et al.*, "Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, pp. 1571–1576.

[43] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *eccv*, 2014.

[44] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," *iccv*, 2013.

[45] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *NIPS Workshop on Deep Learning*, 2014.

[46] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," *ArXiv*, 2018.

[47] L. Xing, Z. Tian, W. Huang, and M. R. Scott, "Convolutional character networks," in *iccv*, 2019.

[48] D. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," 1986.

[49] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask textspotter v3: Segmentation proposal network for robust scene text spotting," in *eccv*, 2020.

[50] W.-Y. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, "Gtc: Guided training of ctc towards efficient and accurate scene text recognition," *ArXiv*, 2020.

[51] R. Litman, O. Anschel, S. Tsiper, R. Litman, S. Mazor, and R. Manmatha, "Scatter: Selective context attentional scene text recognizer," *cvpr*, 2020.

[52] T. Wang *et al.*, "Decoupled attention network for text recognition," *ArXiv*, 2020.

[53] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[54] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Asian conference on computer vision*, Springer, 2014, pp. 35–48.

[55] W. Liu, C. Chen, K.-Y. K. Wong, Z. Su, and J. Han, "Star-net: A spatial attention residue network for scene text recognition.," in *BMVC*, vol. 2, 2016, p. 7.

[56] R. Litman, O. Anschel, S. Tsiper, R. Litman, S. Mazor, and R Manmatha, "Scatter: Selective context attentional scene text recognizer," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 962–11 972.

[57] W. Liu, C. Chen, and K.-Y. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[58] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *nips*, 2014.

[59] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *aaai*, 2016.

[60] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[61] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7094–7103, 2021.

[62] C. Da, P. Wang, and C. Yao, "Levenshtein ocr," *ArXiv*, vol. abs/2209.03594, 2022.

[63] Y. Wang, H. Xie, S. Fang, J. Wang, S. Zhu, and Y. Zhang, "From two to one: A new scene text recognizer with visual language modeling network," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14 174–14 183, 2021.

[64] G. Hinton, O. Vinyals, J. Dean, *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[65] S. Fang, Z. Mao, H. Xie, Y. Wang, C. C. Yan, and Y. Zhang, "Abinet++: Autonomous, bidirectional and iterative language modeling for scene text spotting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 7123–7141, 2022.

[66] Q. Gao, S. Yang, J. Chai, and L. Vanderwende, "What action causes this? towards naive physical action-effect prediction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 934–945.

[67] Q. Gao, M. Doering, S. Yang, and J. Chai, "Physical causality of action verbs in grounded language understanding," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1814–1824.

[68] J. Bi, N. Nguyen, A. Vosoughi, and C. Xu, "MISAR: A multimodal instructional system with augmented reality," *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshop on AV4D: Visual Learning of Sounds in Spaces*, 2023.

[69] A. Padmakumar, M. Inan, S. Gella, P. L. Lange, and D. Hakkani-Tur, "Multimodal embodied plan prediction augmented with synthetic embodied dialogue," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 6114–6131.

[70] G. Sarch, Y. Wu, M. J. Tarr, and K. Fragkiadaki, "Open-ended instructable embodied agents with memory-augmented large language models," *arXiv preprint arXiv:2310.15127*, 2023.

[71] J. Merullo, D. Ebert, C. Eickhoff, and E. Pavlick, "Pretraining on interactions for learning grounded affordance representations," *arXiv preprint arXiv:2207.02272*, 2022.

[72] H. Le, N. F. Chen, and S. C. Hoi, "Multimodal dialogue state tracking," *arXiv preprint arXiv:2206.07898*, 2022.

[73] T. Ates *et al.*, "Craft: A benchmark for causal reasoning about forces and interactions," *arXiv preprint arXiv:2012.04293*, 2020.

[74] T.-L. Wu, Y. Zhou, and N. Peng, "Localizing active objects from egocentric vision with symbolic world knowledge," *arXiv preprint arXiv:2310.15066*, 2023.

[75] R. Zellers *et al.*, "Piglet: Language grounding through neuro-symbolic interaction in a 3d world," *arXiv preprint arXiv:2106.00188*, 2021.

[76] T. Nagarajan and K. Grauman, "Attributes as operators: Factorizing unseen attribute-object compositions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 169–185.

[77] Z. Xue, K. Ashutosh, and K. Grauman, "Learning object state changes in videos: An open-world perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[78] J. Bi, J. Luo, and C. Xu, "Procedure planning in instructional videos via contextual modeling and model-based policy learning," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2021.

[79] Y. Du *et al.*, "Learning universal policies via text-guided video generation," *arXiv preprint arXiv:2302.00111*, 2023.

[80] Y. Zhong, L. Yu, Y. Bai, S. Li, X. Yan, and Y. Li, "Learning procedure-aware video representation from instructional videos and their narrations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 825–14 835.

[81] Y. Tang, J. Zhang, X. Wang, T. Wang, and F. Zheng, "Llmva-gebc: Large language model with video adapter for generic event boundary captioning," *arXiv preprint arXiv:2306.10354*, 2023.

[82] J. Wang, S. Dasari, M. K. Srirama, S. Tulsiani, and A. Gupta, "Manipulate by seeing: Creating manipulation controllers from pre-trained representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3859–3868.

[83] L. Song, J. B. C. Huang, and C. Xu, "Audio-visual action prediction with soft-boundary in egocentric videos,"

[84] Y. Liu, P. Wei, and S.-C. Zhu, "Jointly recognizing object fluents and tasks in egocentric videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2924–2932.

[85] M. F. Naeem, Y. Xian, F. Tombari, and Z. Akata, "Learning graph embeddings for compositional zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 953–962.

[86] N. Saini *et al.*, "Chop & learn: Recognizing and generating object-state compositions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 247–20 258.

[87] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *NeurIPS*, 2022.

[88] H. Touvron *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[89] W.-L. Chiang *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 30 March 2024)*, 2023.

[90] H. W. Chung *et al.*, "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[91] A. Zhang *et al.*, "Transfer visual prompt generator across llms," vol. abs/23045.01278, 2023.

[92] Q. Ye *et al.*, "Mplug-owl: Modularization empowers large language models with multimodality," *arXiv preprint arXiv:2304.14178*, 2023.

[93] B. Li, Y. Zhang, L. Chen, J. Wang, J. Yang, and Z. Liu, "Otter: A multi-modal model with in-context instruction tuning," *arXiv preprint arXiv:2305.03726*, 2023.

[94] P. Gao *et al.*, "Llama-adapter v2: Parameter-efficient visual instruction model," *arXiv preprint arXiv:2304.15010*, 2023.

[95] Z. Peng *et al.*, "Kosmos-2: Grounding multimodal large language models to the world," *arXiv preprint arXiv:2306.14824*, 2023.

[96] Y. Tang *et al.*, *Video understanding with large language models: A survey*, 2023. arXiv: 2312.17432 [cs.CV].

[97] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning*, 2023.

[98] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[99] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.

[100] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *ArXiv*, vol. abs/2304.10592, 2023.

[101] K. Chen, Z. Zhang, W. Zeng, R. Zhang, F. Zhu, and R. Zhao, "Shikra: Unleashing multimodal llm's referential dialogue magic," *ArXiv*, vol. abs/2306.15195, 2023.

[102] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *ArXiv*, vol. abs/2304.08485, 2023.

[103] K. Li *et al.*, "Videochat: Chat-centric video understanding," *ArXiv*, vol. abs/2305.06355, 2023.

[104] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[105] Y. Zhao, I. Misra, P. Krahenbuhl, and R. Girdhar, "Learning video representations from large language models," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6586–6597, 2022.

[106] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," *ArXiv*, vol. abs/2306.02858, 2023.

[107] A. Agrawal *et al.*, "Vqa: Visual question answering," *International Journal of Computer Vision*, vol. 123, pp. 4 –31, 2015.

[108] N. Nguyen, Y. Tian, and C. Xu, "Efficiently leveraging linguistic priors for scene text spotting," *ArXiv*, vol. abs/2402.17134, 2024.

[109] J. Li, D. Li, C. Xiong, and S. C. H. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning*, 2022.

[110] C.-K. Chng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 935–942, 2017.

[111] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognit.*, vol. 90, pp. 337–345, 2019.

[112] D. Karatzas *et al.*, "Icdar 2015 competition on robust reading," *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1156–1160, 2015.

[113] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. J'egou, "Training data-efficient image transformers & distillation through attention," in *ICML*, 2021.

[114] J. Clark, D. Garrette, I. Turc, and J. Wieting, "Canine: Pre-training an efficient tokenization-free encoder for language representation," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 73–91, 2022.

[115] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2315–2324, 2016.

[116] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[117] A. Mishra, A. Karteek, and C. V. Jawahar, "Scene text recognition using higher order language priors," in *British Machine Vision Conference*, 2009.

[118] K. Wang, B. Babenko, and S. J. Belongie, "End-to-end scene text recognition," *2011 International Conference on Computer Vision*, pp. 1457–1464, 2011.

[119] D. Karatzas *et al.*, "Icdar 2013 robust reading competition," *2013 12th International Conference on Document Analysis and Recognition*, pp. 1484–1493, 2013.

[120] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan, "Recognizing text with perspective distortion in natural scenes," *2013 IEEE International Conference on Computer Vision*, pp. 569–576, 2013.

[121] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Syst. Appl.*, vol. 41, pp. 8027–8048, 2014.

[122] N. Nayef *et al.*, "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt," *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 1454–1459, 2017.

[123] A. Singh, G. Pang, M. Toh, J. Huang, W. Galuba, and T. Hassner, "Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8798–8808, 2021.

[124] L. Xing, Z. Tian, W. Huang, and M. R. Scott, "Convolutional character networks," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9125–9135, 2019.

[125] M. Huang *et al.*, "Swintextspotter: Scene text spotting via better synergy between text detection and text recognition," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4583–4593, 2022.

[126] W. Wang *et al.*, "Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 5349–5367, 2021.

[127] R. Ronen, S. Tsiper, O. Anschel, I. Lavi, A. Markovitz, and R. Manmatha, "Glass: Global to local attention for scene-text spotting," in *European Conference on Computer Vision*, 2022.

[128] Y. Kittenplon, I. Lavi, S. Fogel, Y. Bar, R Manmatha, and P. Perona, "Towards weakly-supervised text spotting using a multi-task transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4604–4613.

[129] Y. Baek *et al.*, "Character region attention for text spotting," *ArXiv*, vol. abs/2007.09629, 2020.

[130] D. Peng *et al.*, "Spts: Single-point text spotting," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4272–4281.

[131] X. Zhang, Y. Su, S. Tripathi, and Z. Tu, "Text spotting transformers," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9509–9518, 2022.

[132] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask textspotter v3: Segmentation proposal network for robust scene text spotting," *ArXiv*, vol. abs/2007.09482, 2020.

[133] Y. Liu *et al.*, "Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 8048–8064, 2022.

[134] L. Qiao *et al.*, "Mango: A mask attention guided one-stage scene text spotter," in *AAAI*, 2021.

[135] M. Ye *et al.*, "Deepsolo: Let transformer decoder with explicit points solo for text spotting," *ArXiv*, vol. abs/2305.19957, 2022.

[136] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2035–2048, 2019.

[137] T. Wang *et al.*, "Decoupled attention network for text recognition," in *AAAI Conference on Artificial Intelligence*, 2019.

[138] X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang, "Robustscanner: Dynamically enhancing positional clues for robust text recognition," in *European Conference on Computer Vision*, 2020.

[139] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," *ArXiv*, vol. abs/1811.00751, 2018.

[140] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "Seed: Semantics enhanced encoder-decoder framework for scene text recognition," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 525–13 534, 2020.

[141] Y. He *et al.*, "Visual semantics allow for textual reasoning better in scene text recognition," in *AAAI Conference on Artificial Intelligence*, 2021.

[142] T. Guan *et al.*, "Self-supervised implicit glyph attention for text recognition," 2022.

[143] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Textdragon: An end-to-end framework for arbitrary shaped text spotting," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9075–9084, 2019.

[144] L. Qiao *et al.*, "Text perceptron: Towards end-to-end arbitrary-shaped text spotting," in *AAAI Conference on Artificial Intelligence*, 2020.