

Critical Infrastructure Asset Imaging Pipeline

Jonathan Dupuis

*Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
JonathanDupuis@cmail.carleton.ca*

Dr. James Green

*Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
jrgreen@sce.carleton.ca*

Abstract—The ability to retrieve and analyze recent images of critical infrastructure assets is beneficial for regular monitoring, post-disaster assessment, or preparing for a service call. Given a high-quality image of an asset, several recently developed deep learning models can automatically assess the state of the infrastructure. However, obtaining such an image automatically remains an open question. Enterprise imaging initiatives, such as Google Street View, permit the viewing of road-adjacent images, given geographic coordinates. The spatial resolution of such systems is excellent, although the temporal resolution varies from months to years. We have recently forecast the emergence of on-demand imaging using instrumented vehicles that would permit more recent or frequent imaging of locations of interest. However, the challenge remains to retrieve a high-quality image of an asset of interest, free from obstructions and imaging artifacts. We here propose a pipeline to retrieve recent images of an asset given an imaging source, GPS coordinates, and an asset class. Object detection is used to automatically identify the asset of interest and to detect obstructions or imaging artifacts. If necessary, additional images are requested for surrounding locations to provide multiple views of the asset of interest culminating in an image free from artifacts. The pipeline is demonstrated using two critical infrastructure asset classes (utility poles and street lights) and two image sources (Streetview and a repository of dashcam video). Robust performance is observed, resulting in correct asset identification and imaging in 76.5% of cases (up from 54.5%), while requiring an average of 1.47 images per asset to achieve a high-quality image free from obstructions and artifacts. The proposed pipeline will be of interest to disaster response teams, utilities, and other critical infrastructure asset managers.

Index Terms—deep learning, convolutional neural network, image classification, critical infrastructure monitoring

I. INTRODUCTION

Globally, critical infrastructure is acknowledged for its importance to a prosperous and secure society. In Canada, critical infrastructure is defined as essential processes, systems, facilities, technologies, networks, assets, and services that are crucial to the health, safety, security, economic well-being, and effective functioning of the government [1]. Examples of critical infrastructure assets include fire hydrants, communication equipment, and utility grid infrastructure. Although costly, periodic inspection of such infrastructure is important due to threats, being both acute (e.g., storm damage) or chronic (e.g., wear and tear, vegetation encroachment) in nature. There is ongoing research on using various techniques to analyze and model the resilience of critical infrastructure [2], [3], [4] [5]. Ryan et al [2] highlight the need to consider maintenance

strategies in resilience analysis. MacKenzie and Zobel [6] showed the high cost-effectiveness of advanced monitoring.

Vehicle-based imagery has great potential for monitoring road-adjacent critical infrastructure such as utility poles and fire hydrants. Enterprise-level systematic imaging by Google, Apple, and others has created large repositories of vehicle-based imagery with broad coverage and high spatial resolution. Research has demonstrated the possibility of leveraging existing fleets of municipal vehicles as sensing platforms [7]. As an increasing number of consumer vehicles are equipped with advanced sensing hardware there is also potential for a secondary market for such data, including geotagged images collected on demand [8]. Relative to aerial drone inspection, ground vehicle-based imaging is potentially safer, less expensive, and more frequent [9]. While there is work on sensing issues through a variety of other means (e.g., using smart grids for monitoring electrical infrastructure [10]) visual inspection is still key for monitoring the status of infrastructure assets and diagnosing potential problems, such as ice loading, snow cover, and vandalism. Assessing infrastructure following natural disasters, such as fires, flooding, earthquakes, etc., would also benefit from on-site imaging [11]. Lastly, inspection of road-adjacent infrastructure for unauthorized addition or modification of equipment is another important use case for vehicle-based imagery.

Deep learning has shown promise for automated inspection of critical infrastructure [12], [5]. Models have been developed for automated detection, identification, and segmentation of infrastructure elements, such as segmenting the individual elements that can comprise a power utility pole [13], [9], [14], [15]. Deep learning is also being used for the detection and segmentation of defects in critical infrastructure including segmentation of cracks in concrete structures [16], [17], [18], or measurement of the inclination of utility poles [15].

Given a high-quality image of an asset several research groups have developed computer vision models to automate assessment; however, the question of how to obtain a suitable image free of obstructions and image artifacts remains open. For these models to be deployed by an infrastructure manager, dedicated imaging teams would have to be dispatched to the asset's location, thereby negating much of the efficiency promised by automated infrastructure monitoring. We here propose an image acquisition pipeline that leverages both computer vision and a non-specific source of geotagged im-

agery. Rather than deploying a vehicle specifically to image an asset, we here propose to leverage non-specific image datasets, where images can be requested for a specific GPS coordinate. This image source may be static, such as Google Street View (GSV) [19] or a collection of dashcam footage from a geographic area, or it may be dynamic, where a request is made to a service that can supply an image at the specified location from a passing fleet vehicle suitably equipped with sensors [8].

The challenge of using such pre-existing non-specific image sources is that they may present various imaging artifacts or issues that could impact the ability of a downstream model to effectively leverage the images for inspection or other tasks. The collected images must be free of defects, obstructions, and other artifacts. These potential image issues are not predictable and often appear in a sporadic or dynamic fashion.

For example, GSV is a popular source of street-level infrastructure images [20], [21], [22] due to its extensive coverage. GSV images, however, often contain artifacts that can obscure or warp the subject. Some of these are caused by the GSV panorama stitching process, such as the doubling of objects and blurring of image sections, others are artifacts introduced when the images were taken, like glare or cropping. Examples of these sorts of artifacts are presented in Fig. 3 and further details are presented in section II-B1. Fleet dashcam video has the potential for greater temporal resolution in both recency and frequency. However, it suffers from windshield issues and weather. Lastly, all ground-vehicle-based imaging is prone to obstructions. These issues impacting image quality must be detected and addressed.

We here develop an image collection pipeline that accounts for the possibility of artifacts in the data source. Given the known GPS coordinates of a critical infrastructure asset, the pipeline returns a high-quality image of the asset free from artifacts and cropping. The pipeline automatically detects and corrects various imaging issues by changing the image acquisition location and/or time. This pipeline is able to determine whether images are impacted by such issues and accordingly, take the appropriate steps to obtain a higher quality image. To demonstrate the flexibility of the pipeline, two image sources are examined. The main source of images for our study is GSV, however, we also leverage a dataset of image frames extracted from dashcam videos from an urban environment. Furthermore, two infrastructure asset classes are examined including utility poles and light standards.

This paper makes the following research contributions: 1) We have developed an image acquisition pipeline to provide artifact-free images of a specified infrastructure asset at a given location using a non-specific image dataset; 2) We have developed deep learning models to reliably detect problematic imaging artifacts typical of non-specific image sources; 3) We have compared fine-tuned deep learning models to promptable zero-shot models for these tasks; and 4) We have developed an algorithm to automatically adjust the image location and request new images when artifacts are detected.

II. METHODOLOGY

A. The Pipeline

In order to manage the possible presence of the artifacts mentioned previously, we have developed the pipeline outlined in Fig. 1. At its most basic level, it functions by verifying that there is no overlap between the area of the image containing the asset we are targeting and areas of the image identified as being problematic. When there is an overlap, a new image acquisition location is computed, and the process is repeated.

The pipeline is parameterized by 6 key elements:

- 1) The location of the asset for which an image is required. In our implementation, n locations are defined by geographic coordinates (lat/lon). Our sources for infrastructure asset locations are outlined in section II-B.
- 2) A source of images that can be queried with a location to return either a single image or a batch of multiple images at the given location. In the present study, we have used both GSV and images we collected ourselves as data sources. Our image sources are outlined in section II-B.
- 3) A list of asset classes including the "Asset of Interest" and also classes to be considered "Occluding" classes. The single "Asset of Interest" class is the asset class that the pipeline will attempt to obtain better pictures of. The "Occluding" classes define what is considered problematic elements in the picture for the pipeline to try and avoid.
- 4) An instance segmentation process returning masks and associated class labels for a given image. We implemented multiple deep-learning-based segmentation models, information on these is in section II-C.
- 5) A method for selecting a preferred instance in the case that there are multiple instances of the Asset of Interest class. We here implement a naive selection approach that selects the mask with the greatest area.
- 6) A set of rules or methods dictating how to reposition to query additional pictures from the image source. For the GSV image source, we implemented a movement pattern where we move orthogonal to the original heading towards the target by a set distance. For our dashcam data source, we move backward in time sequentially until a high-quality image is obtained. In both cases, a maximum movement is imposed.

The basic flow of the pipeline, depicted in Fig. 1, is as follows. For a given target asset location, a batch of one or more images is requested from the image source. Depending on the spatial resolution of the image source, and the proximity of the asset to the roadway, the actual image location may differ slightly from the target asset location. Segmentation models then detect and segment (see 4 above), in all images of the batch, all assets for the "Asset of Interest" and "Obstructions" classes (see 3 above). If an asset of interest is found then its mask is checked for overlap with all Occluding class masks in that image. In the case where multiple assets of interest are found, then one is selected and all others are discarded (see 5 above). If there is no overlap, the image is accepted as being of high quality. If there is an overlap, a new location is determined based on a chosen movement method (see 6

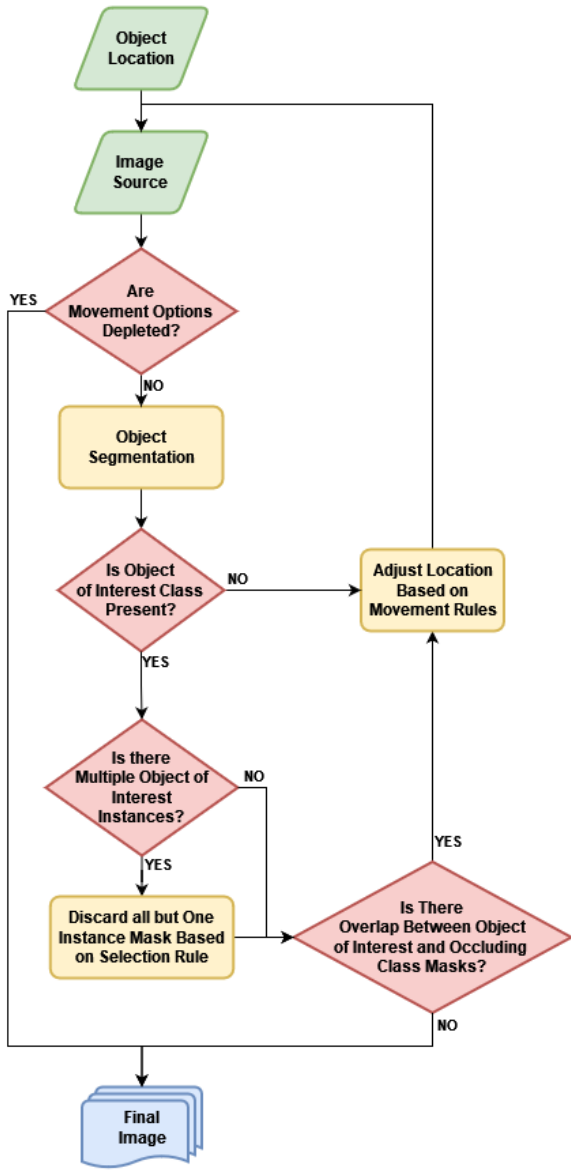


Fig. 1. Pipeline functionality flowchart.

above), and a new image is requested. This process is repeated until the limit of repositioning is reached, at which point the last requested image is returned.

Fig. 2 shows a diagram of an example asset image collection.

Implementers might consider logging cases where no Asset of Interest is found at a given location. This could indicate issues such as unreliable asset coordinates in the dataset, or a bad performing detection model. The meaning and causes of such cases would be implementation specific.

The pipeline allows the composing of different segmentation models to obtain images of a desired asset. This makes it usable by groups or individuals that do not have the ability or resources to train their own models and allows leveraging of foundational models like those listed in section II-C2.

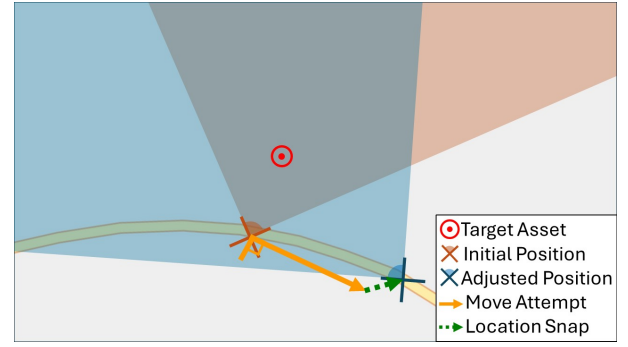


Fig. 2. Overhead view of example asset imaging process with one location adjustment.

B. Datasets

1) *GSV Static Panoramas*: All GSV images were collected through the GSV Static Panoramas API¹. The API requires that a location be present in the requests, either the name of a location as text or latitude and longitude coordinates. The API snaps to the closest available panorama within a 50 meter radius; images returned are therefore not always exactly at the requested location. The heading of the returned image will be in the direction of the requested location; this can be overridden and a heading can be supplied to the request.

Images returned from the API often feature the sorts of artifacts presented in Fig. 3. For privacy reasons, Google will blur sections of GSV imagery [19], in many cases, however, sections of images that do not contain any private imagery are blurred (Fig. 3a). Panoramas are stitched together from multiple images [23], this process is not perfect, and sometimes objects in the images appear duplicated or misaligned (Fig. 3b). In some cases, the direction of the camera aligns with the sun in such a way that the desired subject is partly obscured by glare (Fig. 3c). In many cases, the closest panorama is too close to the desired asset to be fully imaged within the frame (Fig. 3d).

2) *Surrey GSV Dataset*: The city of Surrey British Columbia, makes available an open dataset containing information on 74 165 unique poles of varying types located throughout the city [24].

Of these, 26 690 are utility poles. Each pole includes its geographic coordinates, which were then used to make requests to the GSV static panoramas API. From these panoramas, four 640x640 images that comprise a 360-degree view are extracted. Of the utility poles contained in the Surrey dataset, a subset of 1 000 was randomly chosen, and the 4 000 corresponding images were downloaded from GSV.

All panoramas that did not contain any pole in any of the four pictures were removed from the dataset. The majority of these cases are due to how GSV functions; some API calls, when snapping to their closest panoramas, returned pictures from side streets or pedestrian photospheres. 3 456 images remained after pruning. Examples illustrating the variety of images present in the dataset can be seen in Fig. 4.

¹<https://developers.google.com/maps/documentation/streetview/overview>

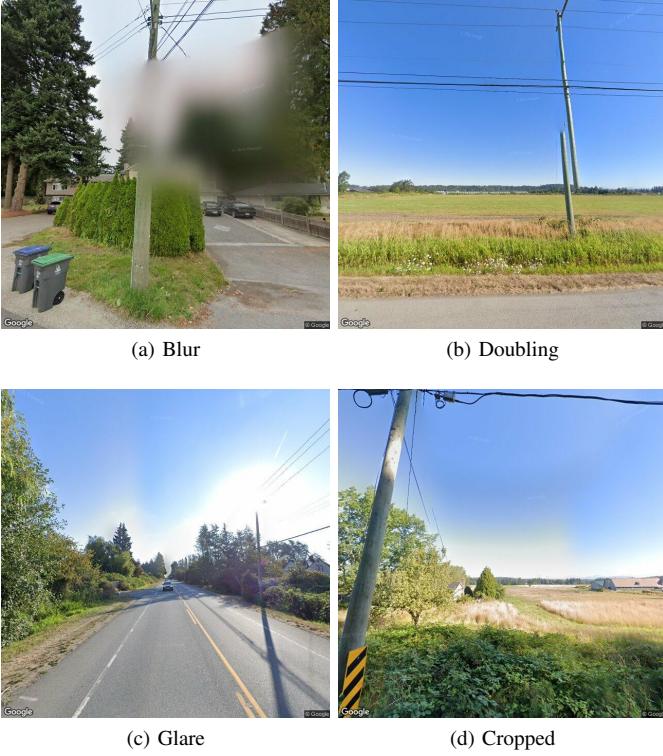


Fig. 3. Sample images from the Surrey GSV dataset showing common artifacts.

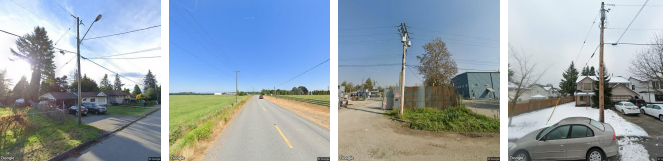


Fig. 4. Sample images from the Surrey dataset.

The images were annotated with masks for five distinct classes.

Three of these are classes representing the artifacts outlined in Section II-B1:

- **Blur Artifact:** Area of image with significant blurring making objects in that region indiscernible, see example in Fig. 3a.
- **Doubling Artifact:** Whole, or part of, object appearing twice or disconnected, see example in Fig. 3b.
- **Glare:** Glare in the image caused by the camera facing the sun directly, see example in Fig. 3c.

The other two classes represent critical infrastructure asset classes:

- **Utility Pole:** A standard utility pole, see examples in Fig. 4.
- **Street Light:** A pole and lamp utilized only for street lighting.

The image dataset is split into training, validation, and testing sets following a random 70%/20%/10% split, resulting in 2419/691/346 images respectively.

TABLE I
TABLE OF INSTANCE COUNTS PER CLASS IN TOTAL AND FOR EACH SPLIT OF THE SURREY DATASET

Class	Total Instances	Train Instances	Val Instances	Test Instances
Assets				
Utility Pole	4287	2976	901	410
Street Light	1364	955	271	138
Artifacts				
Blur Artifact	175	124	38	13
Doubling Artifact	166	106	37	23
Glare	354	252	68	34

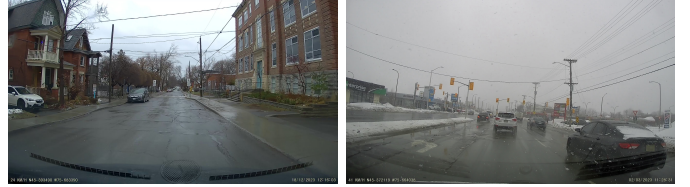


Fig. 5. Sample images from the Ottawa dataset.

3) *Ottawa Dashcam Dataset:* To demonstrate the flexibility of the pipeline, a second geotagged image dataset was collected from dashcam footage taken in Ottawa, Ontario. The footage is all front-facing and taken at a resolution of 2048x1080. 1 hour 17 minutes and 55 seconds of footage was taken across 4 different days, with varying weather conditions. Some of the footage was taken on overlapping routes but most was not. An image frame was extracted at every second of video along with the corresponding GPS coordinates, for a total of 4675 images. Example frames can be seen in Fig. 5.

The city of Ottawa makes available an open dataset containing the location of infrastructure maintained by the municipality. The coordinates from this dataset are used by the pipeline to find the location of utility poles and street lights.

C. Instance Segmentation Models Studied

Asset and artifact class detection and segmentation was done using the models described in the following sub-sections.

1) *YOLO Models:* The YOLO family of models [25] are popular convolutional neural network models for image-related tasks such as classification, detection, and segmentation. They've seen success in applications across many fields such as agriculture, healthcare, remote sensing, and autonomous vehicles [26]. The YOLO models are implemented in several computer vision toolsets, like Roboflow [27], offering easy integration with their platform.

The latest YOLO iteration is the YOLOv8 set of models which has variations for four different tasks: classification, detection, instance segmentation, and pose estimation. There are variants of five different sizes with increasing parameter counts: nano, small, medium, large, and extra large.

Ultralytics makes available checkpoints of model weights pre-trained on the COCO dataset for each segmentation model [25]. We further fine-tuned YOLOv8 models of each size by training on our Surrey dataset for 150 epochs. All models were trained in the same manner, using the AdamW optimizer with

TABLE II

DETECTION AND SEGMENTATION PERFORMANCE ON THE SURREY DATASET TEST SPLIT. BEST PERFORMANCE PER METRIC SHOWN IN BOLD.

Model	Asset Class		Artifact Class		
	Utility Pole	Street Light	Blur	Doubling	Glare
	Box mAP50				
YOLOv8n	79.6	87.0	62.7	29.7	62.6
YOLOv8s	81.1	87.2	68.7	15.1	65.1
YOLOv8m	82.9	87.2	71.1	25.4	68.2
YOLOv8l	82.6	87.0	69.1	25.3	63.5
YOLOv8x	84.8	87.3	63.6	20.1	65.2
GroundedSAM	52.7	-	-	-	-
	Mask mAP50				
YOLOv8n	74.4	73.7	68.5	8.57	61.7
YOLOv8s	73.8	69.8	61.5	11.7	65.3
YOLOv8m	75.8	68.3	72.0	14.8	67.9
YOLOv8l	76.5	68.6	69.4	6.07	63.5
YOLOv8x	76.8	74.6	63.3	2.88	65.2
GroundedSAM	63.1	-	-	-	-

a learning rate of 7.1410×10^{-4} , momentum of 0.9, and a batch size of 16. The training schedule comprised 3 epochs of linear warm-up followed by a cosine learning rate decay down to one percent of the starting learning rate. Mosaic data augmentation was applied for all epochs until the final 10.

The performance metrics on the held-out test set after training are outlined in Table II.

2) *Zero-shot Foundational Models*: Training a performant model is not always a trivial task. Obtaining good quality data can be difficult and labor-intensive. Deep neural networks are also very dependent on compute resources for training, with many popular models having parameter counts too high to be reasonably trainable on consumer hardware.

For these reasons, there has been a trend towards the use of foundational models [28]. These are large models with emergent capabilities that can adapt to new tasks with very little fine-tuning or, in some cases, none at all. The ability of a model to perform new tasks without training is known as zero-shot learning.

We used a combination of two promptable models to achieve zero-shot instance segmentation. GroundedDINO [29] is a multi-modal model that can detect objects in images from open vocabulary text prompts. The resulting bounding boxes can then be used to prompt the Segment Anything Model (SAM) [30], an image segmentation model for general tasks, to obtain the segmentation mask of the object. This combination of methods will be referred to as GroundedSAM.

Different possible text prompts were tested to find the one producing the most accurate object detections. Different backbones for both GroundingDINO and SAM were also tested. These tests were done on the same test splits of the Surrey images as those on which the YOLO models were tested. The asset detection and segmentation results for GroundingDINO are shown in Table III. It can be seen that the larger backbones produce more accurate predictions, and that "utility pole" is a better prompt than potential synonyms like "electricity pole" or "hydro pole".

TABLE III

GROUNDINGDINO + SAM UTILITY POLE DETECTION AND SEGMENTATION PERFORMANCE ON THE SURREY DATASET TEST SPLIT. BEST PERFORMANCE PER METRIC SHOWN IN BOLD.

Grounding-DINO Backbone	SAM Backbone	Prompt	Box mAP50	Mask mAP50
Swin-T	ViT-H	"utility pole"	52.4	57.1
		"electricity pole"	51.1	54.8
		"hydro pole"	42.6	45.2
Swin-T	ViT-B	"utility pole"	52.4	55.5
		"electricity pole"	51.1	54.0
		"hydro pole"	42.6	45.6
Swin-B	ViT-B	"utility pole"	52.7	60.7
		"electricity pole"	50.8	59.5
		"hydro pole"	41.4	47.8
Swin-B	ViT-H	"utility pole"	52.7	63.1
		"electricity pole"	50.8	61.0
		"hydro pole"	41.4	47.8

D. Cropping

Due to the pipeline acting on the presence of overlapping object masks, it is easy to add functionality checking whether the Asset of Interest is cropped by the image border. This can be done by giving to the pipeline, as an occluding element, a mask over a set margin around the picture. This margin can be set to whatever arbitrary ratio of the image one would desire the asset of interest to be contained within.

E. Evaluation and Interpretation of the Pipeline

In order to determine the utility and efficacy of the pipeline the following methodology was used.

Several test samples are taken from a given data source to do a qualitative comparison between the images obtained with the pipeline's intervention and those obtained without. Images obtained from the pipeline are rated as having either an improved, same, or worse quality compared to the images without using the pipeline (i.e., the initial image returned from the geotagged dataset when queried with the asset's location).

The quality judgment is based on the Asset of Interest being cropped and the presence of "blur", "doubling", and "glare" artifacts; and only if they are occluding the asset in the image. For example, in the case of the Surrey dataset, blurring present in an image that does not obscure the main pole would not lower the quality rating, but if any part of the pole is covered by the blurring then that would be considered in the rating. If the asset completely disappears, however, that is then considered a worse outcome. An image is considered improved if there are fewer occluding elements, the same if there is the same amount of occluding elements, or worse if there are more occluding elements or the severity of an occluding element is much greater than in the original image.

There are cases where the target coordinates are erroneous and no asset of interest exists at that location. During the rating process they are noted as such, differentiating them from cases where the model failed to correctly detect the asset.

Once this rating is done for all test samples, comparative measures can be computed showing how often the pipeline took action and how often it led to an improved outcome.

III. RESULTS AND DISCUSSION

The results presented in this section were obtained following the methodology outlined in section II-E.

200 new utility pole coordinates, that did not correspond to any used to build the training dataset, were taken from the Surrey database. The pipeline’s Asset of Interest was set to “utility pole” and the occluding classes were set to “margin”, “blur”, “doubling”, and “glare”. Experiments were run with either YOLO models as predictors for all asset and occluding classes, or with GroundedSAM used as utility pole predictor combined with the medium YOLO model as the occluding class predictor. The exception to this is the cropping detection which is obtained by outlining the image border pixels; a 20 pixel margin was used to determine cropping. Images were sourced from GSV.

The pipeline was run on those 200 examples using different YOLO models and confidence thresholds. The medium and nano YOLO models were selected for comparison. The medium model was selected due to it having the best performance at artifact detection and segmentation (shown in Table II) on the Surrey test set, the nano model was selected to observe the trade-off when using a less accurate but more lightweight model. Each model was tested at two prediction confidence thresholds, 0.5 for each model to minimize false positives and to have a common baseline, and the thresholds at which each model achieved the highest F1 score on the Surrey test set: 0.316 for the nano model and 0.298 for the medium model. GroundedSAM was paired with the medium YOLO model, where the YOLO model segmented the artifacts and GroundedSAM did zero-shot segmentation of utility poles using the text prompt “utility pole”.

To demonstrate the generalizability of the pipeline, two additional tests were conducted: 1) the evaluation process was repeated with 50 new “street light” coordinates from the city of Surrey and images sourced from GSV; and 2) with 50 new “utility pole” coordinates taken from the Ottawa dataset and images sourced from our own collected dashcam footage. These tests were conducted with the medium YOLO model with the conservative 0.5 threshold as the predictor for the asset of interest and all artifact classes except margins.

Table IV shows the significant reduction in artifacts present after collecting the images with the aid of the pipeline compared to their high prevalence when collecting images without. In every case, the number of problematic images was cut down by over half, with the best “model and confidence” combination increasing the ratio of “good” Surrey utility pole images from 54.50% to 79.50%. Images where no utility pole was present in the original GSV request are counted separately from other issues since they are often caused by unreliability in the location data, which cannot be remediated by the pipeline.

Figure 6 shows examples of typical images with artifacts affecting the clarity of the target assets when requesting the nearest GSV images. With the pipeline, clear images free of artifacts are obtained.

A more detailed breakdown of the specific artifacts present across all tests is shown in Table V. Here “obstructed” is added

TABLE IV
NUMBER OF IMAGES CONTAINING ARTIFACTS BEFORE AND AFTER PROCESSING BY THE PIPELINE. CASES WHERE NO ASSET IS PRESENT DUE TO INCORRECT GPS COORDINATES ARE REPORTED SEPARATELY.

Model	Conf.	Artifacts	No Artifacts	No Asset
Surrey Utility Poles				
None	-	69 (34.5%)	109 (54.5%)	22 (11.0%)
YOLOv8n	0.316	25 (12.5%)	153 (76.5%)	22 (11.0%)
	0.500	28 (14.0%)	150 (75.0%)	22 (11.0%)
YOLOv8m	0.298	22 (11.0%)	156 (78.0%)	22 (11.0%)
	0.500	25 (12.5%)	153 (76.5%)	22 (11.0%)
GroundedSAM	0.298	19 (9.5%)	159 (79.5%)	22 (11.0%)
Surrey Street Lights				
None	-	26 (52.0%)	21 (42.0%)	3 (6.0%)
YOLOv8m	0.500	12 (24.0%)	35 (70.0%)	3 (6.0%)
Ottawa Utility Poles				
None	-	6 (12.0%)	27 (24.0%)	17 (34.0%)
YOLOv8m	0.500	1 (2.0%)	32 (64.0%)	17 (34.0%)
Ottawa Street Lights				
None	-	16 (32.0%)	26 (52.0%)	8 (16.0%)
YOLOv8m	0.500	10 (20.0%)	32 (64.0%)	8 (16.0%)

as an issue in order to track cases where the pipeline’s action caused the asset of interest to be significantly blocked from view in the selected image.

In Table VI we see how often the pipeline took action in total, and split into cases of artifacts, no artifacts, and no pole present. We see that, in the majority of cases, the pipeline either correctly repositioned to request a new image, or correctly evaluated the original image to not require any repositioning. From these results, one can measure the rate at which the pipeline makes correct decisions, as shown in Table VII. This shows that, for all models tested, the pipeline took the correct action close to 80% of the time. This table also provided the average number of requests made to the image source per infrastructure element.

Selecting the optimal model to pair with the pipeline can be done in two ways. If the absolute number of artifact-free pictures is the main concern then, following the results from Table IV, the GroundedSAM model would be selected. If it is more important to minimize image requests (e.g., to reduce API usage or limit resource usage due to the object detection and segmentation models), then selection would be done based on the Table VII results which show that the YOLOv8 model has the lowest rate of unjustified readjustments. A lower number of image requests means less incurred costs during deployment. This discrepancy is caused by GroundedSAM having a higher false positive rate but, in many cases, a unnecessary repositioning led to a new image that was equally free of artifacts.

In Table VIII we further break down the effect of the pipeline’s repositioning actions on the quality of the resulting images, showing how often the pipeline taking action led to an improvement in the image quality. We can observe that the pipeline taking action very rarely results in lower quality images, with only around 5% of outcomes after an action being worse. For cases where no artifacts are present but the pipeline decided to take unnecessary repositioning actions, the resulting image is of the same quality in the vast majority of cases.

TABLE V
COUNT OF ARTIFACTS BY TYPE IN IMAGES BEFORE AND AFTER PROCESSING BY THE PIPELINE.

Model	Conf.	Artifacts						
		None	Blur	Cropped	Doubling	Glare	Obstructed	No Pole
Surrey Utility Poles								
None	-	109 (54.5%)	0 (0.0%)	59 (29.5%)	19 (9.5%)	4 (2.0%)	0 (0.0%)	22 (11.0%)
YOLOv8n	0.316	153 (76.5%)	1 (0.5%)	10 (5.0%)	8 (4.0%)	6 (3.0%)	3 (1.5%)	22 (11.0%)
	0.500	150 (75.0%)	1 (0.5%)	13 (6.5%)	9 (4.5%)	7 (3.5%)	2 (1.0%)	22 (11.0%)
YOLOv8m	0.298	156 (78.0%)	0 (0.0%)	9 (4.5%)	7 (3.5%)	5 (2.5%)	3 (1.5%)	22 (11.0%)
	0.500	153 (76.5%)	0 (0.0%)	12 (6.0%)	10 (5.0%)	4 (2.0%)	2 (1.0%)	22 (11.0%)
GroundedSAM	0.298	159 (79.5%)	0 (0.0%)	6 (3.0%)	7 (3.5%)	4 (2.0%)	3 (1.5%)	22 (11.0%)
Surrey Street Lights								
None	-	21 (42.0%)	2 (4.0%)	21 (42.0%)	6 (12.0%)	1 (2.0%)	0 (0.0%)	3 (6.0%)
YOLOv8m	0.500	35 (70.0%)	2 (4.0%)	6 (12.0%)	5 (10.0%)	1 (2.0%)	0 (0.0%)	3 (6.0%)
Ottawa Utility Poles								
None	-	27 (54.0%)	1 (2.0%)	5 (10.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	17 (34.0%)
YOLOv8m	0.500	32 (64.0%)	1 (2.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	17 (34.0%)
Ottawa Street Lights								
None	-	26 (52.0%)	0 (0.0%)	16 (32.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	8 (16.0%)
YOLOv8m	0.500	32 (64.0%)	0 (0.0%)	10 (20.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	8 (16.0%)

TABLE VI
COUNT OF INSTANCES WHERE THE PIPELINE TOOK ACTION, SPLIT BY THE PRESENCE OF ARTIFACTS IN THE ORIGINAL IMAGE.

Model	Conf.	Artifacts Present	Action	
			Moved	Didn't Move
Surrey Utility Poles				
YOLOv8n	0.316	Artifacts	56 (31.5%)	13 (7.3%)
		No Artifacts	26 (14.6%)	83 (46.6%)
		Total	82 (46.1%)	96 (53.9%)
	0.500	Artifacts	51 (28.7%)	18 (10.1%)
		No Artifacts	14 (7.8%)	95 (53.4%)
		Total	65 (36.5%)	113 (63.5%)
YOLOv8m	0.298	Artifacts	56 (31.5%)	13 (7.3%)
		No Artifacts	20 (11.2%)	89 (50.0%)
		Total	76 (42.7%)	102 (57.3%)
	0.500	Artifacts	52 (29.2%)	17 (9.6%)
		No Artifacts	13 (7.3%)	96 (53.9%)
		Total	65 (36.5%)	113 (63.5%)
GroundedSAM	0.298	Artifacts	59 (33.2%)	10 (5.6%)
		No Artifacts	22 (12.4%)	87 (48.9%)
		Total	81 (45.5%)	97 (54.5%)
Surrey Street Lights				
YOLOv8m	0.500	Artifacts	18 (38.3%)	8 (17.0%)
		No Artifacts	2 (4.3%)	19 (40.4%)
		Total	20 (42.6%)	27 (57.4%)
Ottawa Utility Poles				
YOLOv8m	0.500	Artifacts	5 (15.2%)	1 (3.0%)
		No Artifacts	5 (15.2%)	22 (66.7%)
		Total	10 (30.3%)	23 (69.7%)
Ottawa Street Lights				
YOLOv8m	0.500	Artifacts	7 (16.7%)	9 (21.4%))
		No Artifacts	3 (7.1%)	23 (54.8%)
		Total	10 (23.8%)	32 (76.2%)

An analysis of incorrect pipeline actions shows a few reoccurring causes. Chief among these is the misidentification of artifacts. As shown in Table II the models perform significantly less well at detecting the various artifact types than in detecting the infrastructure assets themselves. Common cases include featureless sections of sky being confidently predicted as being “blur” artifacts, or “doubling” cases being missed entirely. The infrastructure assets of interest in some cases were also missed by the models, but this was significantly less of a problem with the medium YOLO model or GroundedSAM compared to the

TABLE VII
METRICS QUANTIFYING CORRECT PIPELINE READJUSTMENT DECISIONS. LISTED ARE THE CORRECT DECISION ACCURACY AMONG ALL CASES, AMONG CASES WHERE A READJUSTMENT WAS MADE (POSITIVE PREDICTIVE VALUE), AND AMONG CASES WHERE NO ACTION WAS TAKEN (NEGATIVE PREDICTIVE VALUE). ALSO LISTED IS THE AVERAGE NUMBER OF IMAGE REQUESTS PER ASSET.

Model	Conf.	Correct Decision	Justified Move	Justified Inaction	No of Images
Surrey Utility Poles					
YOLOv8n	0.316	78.10%	68.29%	86.46%	1.610
	0.500	82.02%	78.46%	84.07%	1.475
YOLOv8m	0.298	81.46%	73.67%	87.25%	1.555
	0.500	83.15%	80.00%	84.96%	1.470
GroundedSAM	0.298	82.02%	72.84%	89.69%	1.625
Surrey Street Lights					
YOLOv8m	0.500	78.72%	90.00%	70.37%	1.560
Ottawa Utility Poles					
YOLOv8m	0.500	81.82%	50.00%	95.65%	1.280
Ottawa Street Lights					
YOLOv8m	0.500	71.43%	70.00%	71.88%	1.420

nano YOLO model. In some cases when using GSV as the data source, repositioning failed to reach a new panorama; in those cases, the pipeline was not able to obtain improved images. Another failure mode was the incorrect selection of the primary asset of interest on which to base the repositioning decision when multiple similar assets were clustered together; for example, utility poles in urban scenes are often closely spaced, making selection of the ‘correct’ instance non-trivial.

IV. CONCLUSION

In this work, we have proposed an algorithmic pipeline to facilitate the deployment of deep learning based critical infrastructure inspection methods. Given the increasing availability of computer vision models for inspecting critical infrastructure, this pipeline addresses the question of how to obtain high-quality images of a given infrastructure asset. We show that our method leads to a three-fold drop in problematic images while triggering very few cases of unnecessary repositioning. Experiments across two cities, two image sources, two infrastructure asset classes, and several object

TABLE VIII
COUNT OF QUALITY OUTCOMES WHEN PIPELINE TOOK ACTION SPLIT BY ORIGINAL PRESENCE OF ARTIFACTS. PERCENTAGES ARE BY LINE, HIGHLIGHT RATIO OF QUALITY IMPROVEMENT OR DEGRADATION CONDITIONAL ON THE PRESENCE OF ARTIFACTS IN THE STARTING IMAGE.

Model	Conf.	Artifacts Present	Quality		
			Improvement	Same	Worse
Surrey Utility Poles					
YOLOv8n	0.316	Artifacts	50 (89.29%)	5 (8.93%)	1 (1.79%)
		No Artifacts	0 (0.00%)	22 (84.62%)	4 (15.38%)
		Total	50 (60.98%)	27 (32.93%)	5 (6.10%)
	0.500	Artifacts	46 (90.20%)	4 (7.84%)	1 (1.96%)
		No Artifacts	0 (0.00%)	11 (78.57%)	3 (21.43%)
		Total	46 (70.77%)	15 (23.08%)	4 (6.15%)
YOLOv8m	0.298	Artifacts	51 (91.07%)	4 (7.14%)	1 (1.79%)
		No Artifacts	0 (0.00%)	18 (90.00%)	2 (10.00%)
		Total	51 (66.23%)	23 (29.87%)	3 (3.90%)
	0.500	Artifacts	47 (90.38%)	4 (7.69%)	1 (1.92%)
		No Artifacts	0 (0.00%)	11 (84.62%)	2 (15.38%)
		Total	47 (71.21%)	16 (24.24%)	3 (4.55%)
GroundedSAM	0.298	Artifacts	53 (89.83%)	4 (6.78%)	2 (3.39%)
		No Artifacts	0 (0.00%)	21 (95.45%)	1 (4.55%)
		Total	53 (61.63%)	30 (34.88%)	3 (3.49%)
Surrey Street Lights					
YOLOv8m	0.500	Artifacts	14 (77.78%)	4 (22.22%)	0 (0.00%)
		No Artifacts	0 (0.00%)	2 (100.00%)	0 (0.00%)
		Total	14 (70.00%)	6 (30.00%)	0 (0.00%)
Ottawa Utility Poles					
YOLOv8m	0.500	Artifacts	5 (100.00%)	0 (0.00%)	0 (0.00%)
		No Artifacts	0 (0.00%)	5 (100.00%)	0 (0.00%)
		Total	5 (50.00%)	5 (50.00%)	0 (0.00%)
Ottawa Street Lights					
YOLOv8m	0.500	Artifacts	6 (85.71%)	1 (14.29%)	0 (0.00%)
		No Artifacts	0 (0.00%)	3 (100.00%)	0 (0.00%)
		Total	6 (60.00%)	4 (4.00%)	0 (0.00%)



(a) 'Before' image has top of pole disconnected from main body due to doubling artifact. 'After' image is clear and free of artifacts.



(b) 'Before' image has a doubling artifact on the bottom half of pole and the pole is cropped by the top image boundary preventing possible equipment inspection. 'After' image is clear and free of artifacts.

Fig. 6. Sample images of Surrey utility poles before (left) and after (right) processing by the pipeline.

segmentation backbone models demonstrate that the proposed pipeline generalizes robustly. We also show the potential of utilizing the pipeline in conjunction with foundational models capable of zero-shot object segmentation to obtain quality imagery of asset classes without needing to train specialized models for an asset class.

Future work will be focused on extending the pipeline to imaging other critical infrastructure asset classes that may present different transient obstructions to imaging than utility poles and street lights. For example, surveying fire hydrants will be less affected by glare and instead present issues such as being blocked by parked vehicles, encroaching vegetation, or snow accumulation.

Extension to other data sources will also be implemented. Potential sources include imaging services similar to GSV, such as Mapillary, and further integration with fleet-based vehicle-mounted geotagging cameras.

V. CURRENT IMPLEMENTATION

Our implementation of the pipeline in Python is available at https://github.com/JDups/GSV_pole_pipeline.

ACKNOWLEDGMENTS

We would like to thank Dr. Kevin Dick and Zein Hajj-Ali for useful discussions and technical assistance with this study. We would also like to thank Anerie Patel for helping with annotations.

REFERENCES

- [1] P. S. Canada, "Canada's Critical Infrastructure," May 2020, last Modified: 2022-10-26. [Online]. Available: <https://www.publicsafety.gc.ca/cnt/ntnl-scrtr/crtcl-nfrstrctr/ccci-iec-en.aspx>
- [2] P. C. Ryan, M. G. Stewart, N. Spencer, and Y. Li, "Reliability assessment of power pole infrastructure incorporating deterioration and network maintenance," *Reliability Engineering & System Safety*, vol. 132, pp. 261–273, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0951832014001744>
- [3] R. Rocchetta, "Enhancing the resilience of critical infrastructures: Statistical analysis of power grid spectral clustering and post-contingency vulnerability metrics," *Renewable and Sustainable Energy Reviews*, vol. 159, p. 112185, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1364032122001095>
- [4] Y. Yao, W. Liu, R. Jain, B. Chowdhury, J. Wang, and R. Cox, "Quantitative metrics for grid resilience evaluation and optimization," *IEEE Transactions on Sustainable Energy*, vol. 14, no. 2, pp. 1244–1258, 2023.
- [5] S. A. Argyroudis, S. A. Mitoulis, E. Chatzi, J. W. Baker, I. Brilakis, K. Gkoumas, M. Voudoukas, W. Hynes, S. Carluccio, O. Keou, D. M. Frangopol, and I. Linkov, "Digital technologies can enhance climate resilience of critical infrastructure," *Climate Risk Management*, vol. 35, p. 100387, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212096321001169>
- [6] C. A. MacKenzie and C. W. Zobel, "Allocating resources to enhance resilience, with application to superstorm sandy and an electric utility," *Risk Analysis*, vol. 36, no. 4, pp. 847–862, 2016. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/risa.12479>
- [7] A. Anjomshoa, F. Duarte, D. Rennings, T. J. Matarazzo, P. deSouza, and C. Ratti, "City scanner: Building and scheduling a mobile sensing platform for smart city services," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4567–4579, 2018.
- [8] K. Dick and J. R. Green, "Emergence of an autonomous vehicle secondary data market for breakthrough applications," in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 4909–4915.
- [9] Y. S. Dosso, E. Rizcallah, F. Kwamena, R. Goubran, and J. R. Green, "Deep learning for segmentation of critical electrical infrastructure from vehicle-based images," in *2022 IEEE Electrical Power and Energy Conference (EPEC)*, 2022, pp. 241–247.
- [10] A. Traoré, M. Chetoui, F.-G. Landry, and M. A. Akhloufi, "Ensemble learning framework to detect partial discharges and predict power line faults," in *2021 IEEE Electrical Power and Energy Conference (EPEC)*, 2021, pp. 285–289.
- [11] M.-K. Kim, S. P. Kim, N. H. Kim, Y. K. Song, and H.-G. Sohn, "Image Quality Assessment of Mobile-based Image Acquisition System for Disaster Scientific Investigation," *Journal of Korean Society for Geospatial Information System*, vol. 24, no. 3, pp. 75–83, Sep. 2016, publisher: The Korean Society for Geospatial Information Systems.
- [12] K. Dick, L. Russell, Y. Dosso, F. Kwamena, and J. Green, "Deep learning for critical infrastructure resilience," *Journal of Infrastructure Systems*, vol. 25, 06 2019.
- [13] D. Zünd and L. M. A. Bettencourt, "Street View Imaging for Automated Assessments of Urban Infrastructure and Services," in *Urban Informatics*, W. Shi, M. F. Goodchild, M. Batty, M.-P. Kwan, and A. Zhang, Eds. Singapore: Springer Singapore, 2021, pp. 29–40. [Online]. Available: https://doi.org/10.1007/978-981-15-8983-6_4
- [14] A. Singh, S. Rajan, M. Amini, J. R. Green, and K. Dick, "Critical electrical infrastructure segmentation in arctic conditions," in *2023 IEEE Sensors Applications Symposium (SAS)*, 2023, pp. 01–06.
- [15] L. Chen, J. Chang, J. Xu, and Z. Yang, "Automatic Measurement of Inclination Angle of Utility Poles Using 2D Image and 3D Point Cloud," *Applied Sciences*, vol. 13, no. 3, 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/3/1688>
- [16] D. Kang, S. S. Benipal, D. L. Gopal, and Y.-J. Cha, "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning," *Automation in Construction*, vol. 118, p. 103291, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0926580520300157>
- [17] P. Savino and F. Tondolo, "Civil infrastructure defect assessment using pixel-wise segmentation based on deep learning," *Journal of Civil Structural Health Monitoring*, vol. 13, no. 1, pp. 35–48, Jan. 2023. [Online]. Available: <https://doi.org/10.1007/s13349-022-00618-9>
- [18] Y. Bai, B. Zha, H. Sezen, and A. Yilmaz, "Engineering deep learning methods on automatic detection of damage in infrastructure due to extreme events," *Structural Health Monitoring*, vol. 22, no. 1, pp. 338–352, 2023, eprint: <https://doi.org/10.1177/14759217221083649>. [Online]. Available: <https://doi.org/10.1177/14759217221083649>
- [19] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *IEEE Computer*, vol. 43, pp. 32–38, 06 2010.
- [20] P. Salesses, K. Schechtner, and C. A. Hidalgo, "The collaborative image of the city: Mapping the inequality of urban perception," *PLOS ONE*, vol. 8, no. 7, pp. 1–12, 07 2013. [Online]. Available: <https://doi.org/10.1371/journal.pone.0068400>
- [21] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese, "Generic 3D representation via pose estimation and matching," in *European Conference on Computer Vision*. Springer, 2016, pp. 535–553.
- [22] P. Mirowski, A. Banki-Horvath, K. Anderson, D. Teplyashin, K. M. Hermann, M. Malinowski, M. K. Grimes, K. Simonyan, K. Kavukcuoglu, A. Zisserman *et al.*, "The streetlearn environment and dataset," *arXiv preprint arXiv:1903.01292*, 2019.
- [23] "How Street View works and where we will collect images next." [Online]. Available: <https://www.google.com/streetview/how-it-works/>
- [24] T. O. D. City of Surrey, "Poles: City of surrey open data." 02 2014. [Online]. Available: <https://data.surrey.ca/dataset/poles/resource/b7c9466f-213b-4b3a-9341-3fcc76dcd65a>
- [25] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [26] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023. [Online]. Available: <https://www.mdpi.com/2504-4990/5/4/83>
- [27] J. G. JAN 10 and . . M. Read, "Launch: Deploy YOLOv8 with Roboflow," Jan. 2023. [Online]. Available: <https://blog.roboflow.com/upload-model-weights-yolov8/>
- [28] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kudithipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Muniyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the opportunities and risks of foundation models," 2022.
- [29] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.
- [30] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.