INSTRUCTCV: INSTRUCTION-TUNED TEXT-TO-IMAGE DIFFUSION MODELS AS VISION GENERALISTS

Yulu Gan
Peking University

Sungwoo Park
UC Berkeley

Alexander Schubert
UC Berkeley and UCSF

Anthony Philippakis
Broad Institute of MIT & Harvard

Ahmed M. Alaa
UC Berkeley and UCSF

ABSTRACT

Recent advances in generative diffusion models have enabled text-controlled synthesis of realistic and diverse images with impressive quality. Despite these remarkable advances, the application of text-to-image generative models in computer vision for standard visual recognition tasks remains limited. The current de facto approach for these tasks is to design model architectures and loss functions that are tailored to the task at hand. In this paper, we develop a *unified language interface* for computer vision tasks that abstracts away task-specific design choices and enables task execution by following natural language instructions. Our approach involves casting multiple computer vision tasks as text-to-image generation problems. Here, the *text* represents an *instruction* describing the task, and the resulting *image* is a visually-encoded *task output*. To train our model, we pool commonly-used computer vision datasets covering a range of tasks, including segmentation, object detection, depth estimation, and classification. We then use a large language model to paraphrase prompt templates that convey the specific tasks to be conducted on each image, and through this process, we create a multi-modal and multi-task training dataset comprising input and output images along with annotated instructions. Following the InstructPix2Pix architecture, we apply instruction-tuning to a text-to-image diffusion model using our constructed dataset, steering its functionality from a generative model to an instruction-guided multi-task vision learner. Experiments demonstrate that our model, dubbed *InstructCV*, performs competitively compared to other generalist and task-specific vision models. Moreover, it exhibits compelling generalization capabilities to unseen data, categories, and user instructions.

Code: https://github.com/AlaaLab/InstructCV

Demo: https://huggingface.co/spaces/alaa-lab/InstructCV

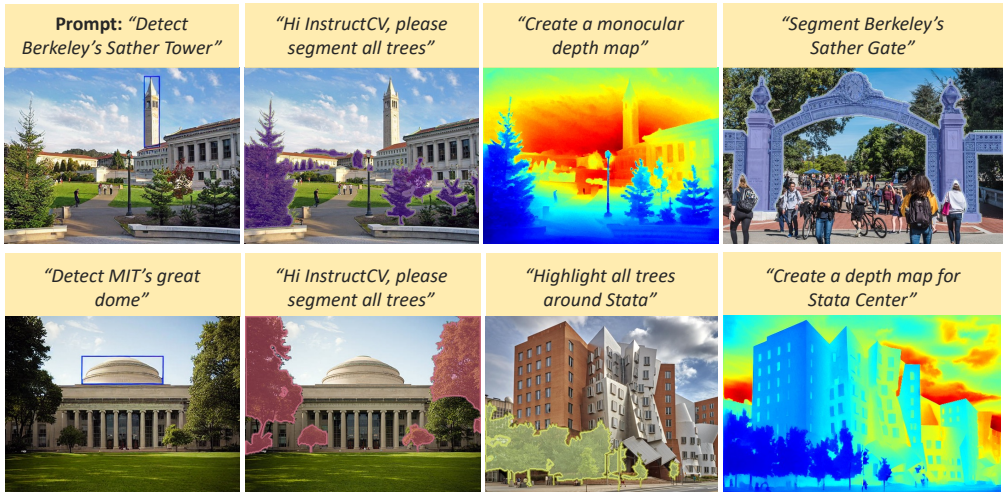

Figure 1: **Application of InstructCV to new test images & user-written instructions:** InstructCV performs the vision task described in the instruction on the input image. *(Images courtesy of UC Berkeley and MIT).*

1 INTRODUCTION

Recent work on text-to-image models has achieved impressive performance in image synthesis [1–3]. Particularly, diffusion models [4–7] have demonstrated remarkable capabilities of transforming diverse text prompts into realistic images, even for novel concepts. Models like DALL·E [2] and Stable Diffusion [8] highlight this progress, now finding use in real-world applications. However, despite these impressive results, generative text-to-image models have so far not been exploited as a unified basis for standard visual recognition tasks. Instead, the predominant approach for these tasks is to design dedicated task-specific architectures and loss functions [9–11], foregoing the opportunity to learn generalizable representations across heterogeneous problem domains and data landscapes.

Previous attempts to create unified models for computer vision tasks have predominantly relied on *prompt tuning* approaches in conjunction with sequence-to-sequence architectures [9–17]. This general framework enables conditioning on input images as well as task-specific prompts by representing image pixels and (trainable) prompts as sequences of discrete tokens. When trained on multi-task datasets, the resulting output tokens align with the desired outcomes for the respective tasks prompted. One illustrative example of this approach is Pix2Seq [16], which follows an autoregressive language modeling approach for processing tokenized image pixels and task identifier codes. Another example includes a class of methods based on *visual prompting*, which defines task-specific prompts in pixel space, unifying multiple vision tasks within a common "inpainting" framework [18, 19]. In both examples, the task-specific prompts steer a single architecture to execute multiple vision tasks. However, these prompts consist of (uninterpretable) numerical values derived from specific training datasets, which may limit their ability to generalize to new datasets, tasks, or categories.

In this paper, we propose a unified model for computer vision tasks that conducts a given task by following natural language instructions (Fig. 1). Our framework, dubbed *InstructCV*, repurposes generative text-to-image models to create a universal language interface for vision tasks. It does so by casting multiple computer vision tasks as text-to-image generation problems, where textual prompts (instructions) serve as explicit task descriptors, guiding the generation process to produce the visual task output corresponding to the input image. By conditioning on natural language descriptions of vision tasks, InstructCV enhances the representation of semantic coherence between images and language prompts, improving the model's generalization capabilities to new human-written instructions and new categories compared to prior "generalist" vision models [12, 13, 15–26].

To train InstructCV, we follow an *instruction tuning* approach applied to a pretrained conditional diffusion model (Stable Diffusion). We generate the instruction tuning data by constructing a multi-modal, multi-task training dataset that comprises tuples of textual instructions, input images and visually-encoded task outputs. We do so by first combining several standard computer vision datasets across multiple tasks including segmentation, object detection, depth estimation, and classification. Next, in order to create heterogeneous and semantically rich textual instructions, we use a large language model (LLM) to paraphrase prompt templates for each vision task. Finally, we encode the output of the vision task associated with each instruction in the form of an output image (e.g., a masking pattern for semantic segmantation). Using this dataset, we utilize the InstructPix2Pix architecture [27] to instruction-tune a text-to-image diffusion model, transforming its functionality from a generative image synthesis model into an instruction-guided multi-task vision learner.

Our experiments demonstrate that InstructCV achieves competitive results compared to other vision generalist and task-specific vision models. Particularly, InstructCV displays compelling generalization properties, surpassing the performance of state-of-the-art vision generalist models on external datasets as well as on unseen prompts in open-vocabulary segmentation tasks.

2 INSTRUCTCV

The InstructCV framework comprises two key steps: (a) construction of a multi-modal and multi-task instruction-tuning dataset, and (b) finetuning a pretrained conditional diffusion model using the dataset generated in step (a). (See Fig. 2 for a pictorial depiction.) Details of each step are provided below.

2.1 CONSTRUCTING A MULTI-MODAL & MULTI-TASK INSTRUCTION-TUNING DATASET

We combine four widely-used computer vision datasets (MS-COCO [28], ADE20K [29, 30], Oxford-III-Pets [31] and NYUv2 [32]) covering four vision tasks (semantic segmentation, object detection,

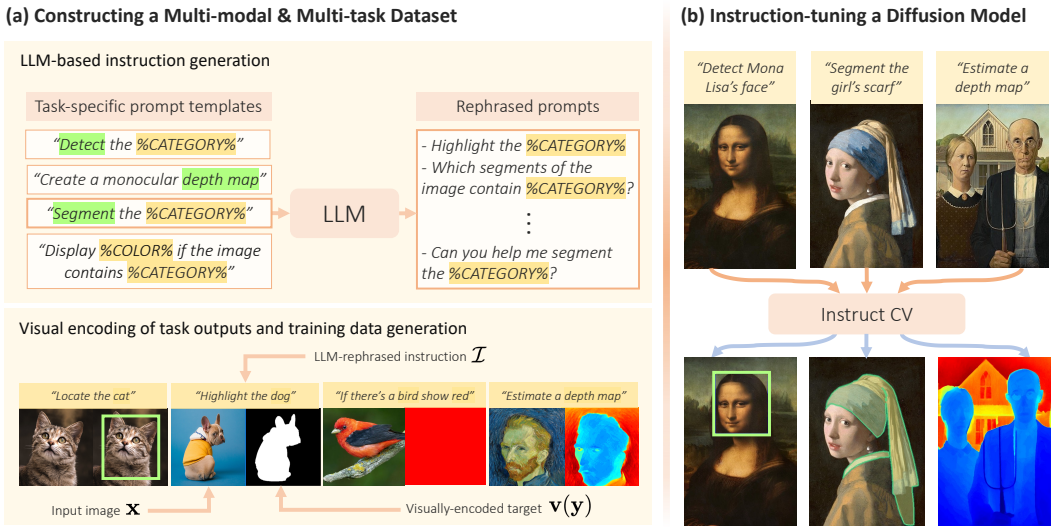

Figure 2: **Pictorial depiction of the InstructCV training pipeline.** (a) We pool multiple computer vision datasets to construct a multi-modal and multi-task set of image pairs, where the target of each task is visually encoded in the form of an output image. Starting with a set of task-specific prompt templates, we sample a new instruction for each training point by using an LLM to rephrase the template for the corresponding task. (b) Using the dataset in (a), we finetune a diffusion model to produce the output $\mathbf{v}(\mathbf{y})$ given an image $\mathbf{x}$ & an instruction $\mathcal{I}$.

monocular depth estimation and classification), into a single multi-task dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i, m_i)\}_i$ in which $\mathbf{x}_i$ is the input image, $\mathbf{y}_i$ is the task output and $m_i \in \{1, \dots, M\}$ is the task identifier ($M = 4$ in our setup). We convert $\mathcal{D}$ into a multi-modal instruction-tuning dataset $\mathcal{D}_\mathcal{I} = \{(\mathbf{x}_i, \mathbf{v}(\mathbf{y}_i), \mathcal{I}_i)\}_i$, in which the task identifier for training point i is expressed as a natural language instruction $\mathcal{I}_i$, and the label $\mathbf{y}_i$ is represented in a visual format $\mathbf{v}(\mathbf{y}_i)$. We construct the dataset $\mathcal{D}_\mathcal{I}$ through the following steps.

LLM-based instruction generation. For each vision task $m \in \{1, \dots, M\}$ under consideration, we pick a prompt template $\mathcal{I}^m_{\text{temp}}$ that describes the task, e.g., *"Segment the %category%"* for the semantic segmentation task. We then attach a prompt to each training data point by inserting the category within the image in the corresponding task template, e.g., $\mathcal{I}^m_{\text{temp}}(\mathbf{x}_i, \mathbf{y}_i)$ = *"Segment the cat"*. We consider two incarnations of our instruction-tuning dataset. First, we consider a baseline dataset that only uses the deterministic task-specific prompt templates described above. We refer to as this dataset as the fixed prompts (FP) instruction-tuning dataset $\mathcal{D}^{\text{FP}}_\mathcal{I} = \{(\mathbf{x}_i, \mathbf{v}(\mathbf{y}_i), \mathcal{I}^{m_i}_{\text{temp}}(\mathbf{x}_i, \mathbf{y}_i))\}_i$. In addition, we use a T5-based paraphrasing LLM [33, 34] to generate rephrased versions of the prompt template to create a diverse range of instructions. As shown in Fig. 2(a) (top), the LLM takes as an input the prompt template (e.g., *"Segment the %category%"*) and produces a wide range of paraphrased variants (e.g., *"Highlight the %category%"*). This procedure ensures that our instruction set is varied yet firmly tied to the core intent of the original prompt. We use the LLM to sample a rephrased variant of the prompt template $\mathcal{I}_i \sim \text{LLM}(\mathcal{I}^{m_i}_{\text{temp}})$ for each training data point i in $\mathcal{D}$. We refer to the resulting instruction tuning dataset as the rephrased prompt (RP) dataset $\mathcal{D}^{\text{RP}}_\mathcal{I} = \{(\mathbf{x}_i, \mathbf{v}(\mathbf{y}_i), \mathcal{I}_i(\mathbf{x}_i, \mathbf{y}_i))\}_i$.

Visual encoding of task outputs. We format the target label $\mathbf{y}$ of each task to represent it in the same RGB image space as the input image $\mathbf{x}$ through a "visual encoding" function $\mathbf{v}(\mathbf{y})$ (Fig. 2(a) (bottom)). This enables casting all tasks in a unified text-to-image generation framework, leveraging Pix2Pix architectures. That is, given an image $\mathbf{x}$ and instruction $\mathcal{I}$, InstructCV produces an image $\mathbf{v}(\mathbf{y})$ that encodes the task output. In the following, we provide the definition of $\mathbf{v}(\mathbf{y})$ for all tasks under study.

(1) Semantic Segmentation. The target output $\mathbf{y}$ of this task is typically an assignment of a label or category to every pixel in an image. A natural choice of $\mathbf{v}(\mathbf{y})$ for the semantic segmentation task is a binary mask that labels pixels in the input image $\mathbf{x}$ belonging to the prompted category in $\mathcal{I}$.

(2) Object Detection. Here, the goal is to identify the spatial position of a category in an image using a bounding box, i.e., the label comprises bounding box coordinates $\mathbf{y} = [c_x, c_y, w, h]$. We define $\mathbf{v}(\mathbf{y})$ for object detection as the image $\mathbf{x}$ with a bounding box overlaid according to the coordinates in $\mathbf{y}$.

(3) Monocular Depth Estimation. The target $\mathbf{y}$ of this task is the depth value (i.e., distance relative to the camera) of each pixel in the RGB image $\mathbf{x}$. For this task, we define the visually-encoded target

$\mathbf{v}(\mathbf{y})$ as an RGB image in which pixel colors encode the depth values. This encoding is done by converting depth values ranging from 0 to 10 meters (based on depth ranges in the NYUv2 dataset [32]) into the discrete space $[0, 1, \ldots, 255]$ for RGB image representation, i.e., $\mathbf{v}(\mathbf{y}) = \lfloor \mathbf{y} \times \frac{255}{10} \rfloor$. We then apply the same value across all three RGB channels to create a visual depth map.

(4) Image Classification. In multi-class image classification, the target label $\mathbf{y}$ is a categorical value indicating the object depicted in the image $\mathbf{x}$. To represent image classification in a Pix2Pix format, we resort to a color-coding methodology. To this end, we use a prompt template of the form: *"Display %color% if the image contains %category%"*. We sample random colors when filling in the template for individual training points. This steers the text-to-image model to produce an image consisting of the pure color block if the category specified in the prompt is visible in $\mathbf{x}$. (Note that this approach only enables us to predict if a specific category is in the input image $\mathbf{x}$. For multi-class classification, we need to use a series of prompts specifying all categories of interest one at a time.)

2.2 Instruction-tuning a Latent Diffusion Model

We use our instruction-tuning dataset $\mathcal{D}_{\mathcal{I}} = \{(\mathbf{x}_i, \mathbf{v}(\mathbf{y}_i), \mathcal{I}_i)\}_i$ to train a (conditional) diffusion model that conducts the vision task specified in the instruction $\mathcal{I}$ on the input image $\mathbf{x}$, producing a visually-encoded task output $\mathbf{v}(\mathbf{y})$. By finetuning the text-to-image diffusion model using $\mathcal{D}_{\mathcal{I}}$, we steer its functionality from a generative model to a language-guided multi-task vision learner. We use a training procedure similar to that of InstructPix2Pix [27], which applies uses a similar multi-modal dataset (pairs of images and editing instructions) to train an instruction-guided image editing model.

Diffusion models [7] generate data by gradually denoising a normally distributed random variable; this process amounts to learning the reverse dynamics of a Markov chain with fixed length T. Latent diffusion models [8] applies this approach within the latent space of a pretrained variational autoencoder [35] with encoder $\boldsymbol{E}(.)$ and decoder $\boldsymbol{D}(.)$. Training a diffusion model involves a forward *diffusion* and a reverse *denoising* process. During the forward process, the image $\mathbf{v}(\mathbf{y})$ is transformed to its latent representation $\mathbf{z}_0 = \boldsymbol{E}(\mathbf{v}(\mathbf{y}))$, which is then injected with Gaussian noise over T steps:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t \mathbf{I}), \ \forall t \in \{1, \ldots, T\}, \tag{1}$$

where the time-varying constants $\beta_{1:T}$ control how much noise is added at each timestep t and are chosen such that z_T roughly converges to a standard Gaussian vector. This forward process does not contain any trainable parameters and can be described as $q(\mathbf{z}_{1:T}|\mathbf{z}_0) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1})$. In the reverse diffusion process, $p_\theta(\mathbf{z}_{0:T}) = p(\mathbf{z}_T) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$ the objective is to learn a model to progressively denoise the latents $\mathbf{z}_{T:1}$ to recover the initial latent z_0. The target image can then be reconstructed as $\mathbf{v}(\mathbf{y}) = \boldsymbol{D}(\mathbf{z}_0)$. The reverse diffusion process can be written as:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, t), \rho_t^2 \mathbf{I}), \ \forall t \in \{1, \ldots, T\}, \tag{2}$$

where the means μ_θ are typically parameterized using neural networks and the variances ρ_t^2 are predetermined constants. Such a denoising model is learned by optimizing a reweighted variant of the variational lower bound on the data distribution [4, 36], i.e.,

$$\mathcal{L}_{\text{unconditional}} := \mathbb{E}_{\boldsymbol{E}(\mathbf{v}(\mathbf{y})), \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), t} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(t, \mathbf{z}_t\right) \right\|_2^2 \right], \tag{3}$$

where $\boldsymbol{\epsilon}$ is a standard normal random variable and $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$, where $\alpha_t^2 = \prod_{s=1}^{t}(1-\beta_s)$ and $\sigma_t^2 = 1 - \alpha_t^2$ are based on the diffusion distribution $q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{z}_0, \sigma_t^2 \mathbf{I})$. The noise predictor $\boldsymbol{\epsilon}_\theta$ is obtained from the parameterization $\mu_\theta(\mathbf{z}_t, t) := (\mathbf{z}_t - \beta_t \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)/\sqrt{1-\alpha_t^2})/\sqrt{1-\beta_t}$. The model $\boldsymbol{\epsilon}_\theta$ is trained to predict the noise vector ϵ at each time-step t in order to denoise the latent variable $\mathbf{z}_t$.

Instruction-tuning via image and text conditioning. The objective in (3) can be further refined to condition on both the input image $\mathbf{x}$ and instruction $\mathcal{I}$ to generate the desired output $\mathbf{v}(\mathbf{y})$—this can be achieved by learning a conditional noise predictor [8, 27], minimizing the following loss function:

$$\mathcal{L}_{\text{conditional}} := \mathbb{E}_{\boldsymbol{E}(\mathbf{v}(\mathbf{y})), \boldsymbol{E}(\mathbf{x}), \mathcal{I}, \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), t} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(t, \mathbf{z}_t, \boldsymbol{E}(\mathbf{x}), \mathcal{I}\right) \right\|^2 \right]. \tag{4}$$

We use a pretrained Stable Diffusion checkpoint, which exhibits strong text-to-image generation capabilities, as the backbone architecture of our model. For text conditioning, we adopt the same methodology as in [8], utilizing the instruction $\mathcal{I}$ instead of image captions as textual inputs. For image conditioning, we concatenate the encoded input image $\mathbf{E}(\mathbf{x})$ with the latent $\mathbf{z}_t$, which are then fed into input to the first layer of the noise predictor ϵ_θ [27].

Classifier-free guidance. To further improve the alignment between the generated outputs and their conditioning, we apply "classifier-free" guidance [37]. This approach combines the unconditional and conditional noise predictors in (4) and (3) in order to shift probability mass towards data where an implicit classifier $p_\theta(c|\mathbf{z}_t)$ assigns a high score to the conditioning c. Similar to InstructPix2Pix ((3) in [27]), we use a modified noise predictor that assigns different weights to the different components of the conditioning $(\mathbf{x}, \mathcal{I})$ as follows:

$$\begin{aligned}\tilde{\epsilon}_\theta\left(t, \mathbf{z}_t, \mathbf{x}, \mathcal{I}\right) &= \epsilon_\theta(t, \mathbf{z}_t, \varnothing, \varnothing) \\ &+ \gamma_I \cdot \left(\epsilon_\theta(t, \mathbf{z}_t, \mathbf{x}, \varnothing) - \epsilon_\theta(t, \mathbf{z}_t, \varnothing, \varnothing)\right) \\ &+ \gamma_T \cdot \left(\epsilon_\theta(t, \mathbf{z}_t, \mathbf{x}, \mathcal{I}) - \epsilon_\theta(t, \mathbf{z}_t, \mathbf{x}, \varnothing)\right).\end{aligned}$$

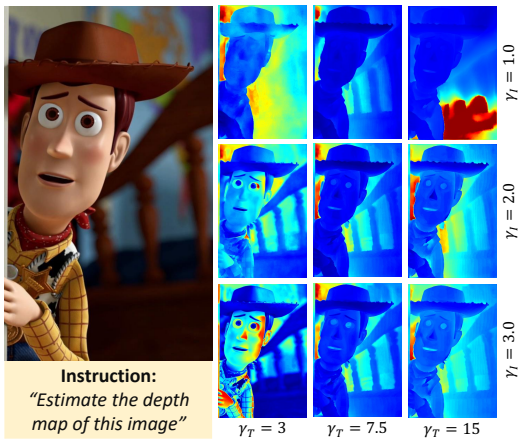

Figure 3: **Impact of classifier-free guidance** on the outputs of InstructCV for the depth estimation task.

where γ_I and γ_T are guidance scales that determine the relative importance of the image and text conditioning. Unconditional denoising is achieved by introducing null values to the respective image and text channels of the noise predictor. In Fig. 3, we show the effects of the parameters γ_I and γ_T on the output image $\mathbf{v}(\mathbf{y})$ of InstructCV. As we can see, γ_I controls the similarity between the input and output images, $\mathbf{x}$ and $\mathbf{v}(\mathbf{y})$, whereas γ_T controls the extent to which pixel values $\mathbf{v}(\mathbf{y})$ correspond to actual depth values in $\mathbf{x}$. Low γ_T and high γ_I result in output images that look very similar to input images with only a modification in the color map that does not reflect pixel depths. On the other hand, high γ_T and low γ_I produces coarse depth maps that miss nuanced details in the input images.

3 Related Work

Repurposing diffusion models for vision tasks. Diffusion models have achieved impressive performance in image generation [4, 36, 38–40], text-to-image synthesis [1, 2, 6, 8], as well as generation of other modalities such as video [41] and audio [42]. The idea of repurposing diffusion models to tackle standard computer vision tasks has been considered before to develop models for object detection [43] and open-vocabulary panoptic segmentation [44]. These approaches were limited to single-task settings with specialized loss functions and (unimodal) architectures. Contrarily, InstructCV provides a unified architecture for multi-task learning, with a natural language interface that enhances generalization to new datasets and categories. The idea of "instruction-tuning" text-to-image diffusion models was introduced in [27] with the objective of finetuning the model to follow editing instructions. InstructCV builds on this framework to adapt text-to-image models for performing conventional visual recognition tasks. To our knowledge, this is one of the earliest efforts in this direction.

Vision Generalists. Several prior attempts have aimed to develop unified models capable of executing multiple vision tasks within a single, shared architecture. Motivated by successes of LLMs, recent work has attempted to design such generalist models based on sequence-to-sequence architectures. Among these, models such as Florence [9], OFA [10], CoCa [11] and BEiT-3 [14], learn general representation encoder, which require individual finetuning to each specific downstream task. Methods such as Unified-IO [15] and Pix2Seq-v2 [17] build single architectures that are capable of performing multiple vision tasks via prompt tuning. However, the sequence-based operation of these models results in slow inference speeds, and the tuned prompts may not generalize to unseen datasets/categories. Another line of work proposes vision transformer-based architectures that frame different vision tasks as inpainting problems [18, 19]. This work focuses on in-context learning based on visual prompts and does not consider language-based instructions, which we believe is a more natural interface for general-purpose models. To the best of our knowledge, the only generalist model that supports a language-based interface for vision tasks similar to that of InstructCV is the VisionLLM model developed in [13]. This is an LLM-based framework that treats images as a foreign language and aligns vision-centric tasks with language tasks that can be flexibly defined using language-based instructions. VisionLLM and InstructCV share a common objective but use different approaches. VisionLLM finetunes a pre-trained LLM using vision-centric tasks, whereas InstructCV finetunes a text-to-image model by substituting image captions with instructional text. We were unable to empirically compare the two models as the code for VisionLLM was not available at the time of writing this paper.

4 EXPERIMENTS

4.1 EXPERIMENTAL SET UP

We evaluate InstructCV across the four vision tasks under study (semantic segmentation, object detection, monocular depth estimation and image classification). For this purpose, we consider widely used datasets for each task: ADE20k [29, 30] for semantic segmentation, MS-COCO [28] for object detection, NYUv2 for depth estimation [45] and Oxford-IIIT Pet [46] for classification. In what follows, we explain the processing steps and evaluation procedure for all tasks under consideration.

Semantic Segmentation. ADE20K covers 150 semantic categories and comprises $25,000$ images of which we use $20,000$ for training, $2,000$ for validation, and $3,000$ for testing. We follow the same protocol as suggested in [18] to implement the training/test split. At inference time, we average the outputs of the three channels of the output image $\mathbf{v}(\mathbf{y})$ to obtain the final segmentation mask. We evaluate the accuracy of segmentation masks using the Mean Intersection over Union (mIoU) metric.

Object Detection. MS-COCO contains $118,000$ training and $5,000$ validation images with labels for 80 different categories. We follow the same protocol as in Pix2Seq [16] to set up the training/test split. At inference time, we follow the post-processing steps in Appendix A.3 to derive the coordinates and category of each Region of Interest (RoI) from the output image $\mathbf{v}(\mathbf{y})$. We then aggregate the results for all categories in order to calculate the Mean Average Precision (mAP).

Depth Estimation. The NYUv2 dataset [32] consists of 464 indoor scenes captured by a Microsoft Kinect camera. We follow the official training/test split, with $24,231$ image-depth pairs used for training, and 654 used for testing. For the test images we report the Root Mean Square Error (RMSE), absolute mean relative error (A.Rel), and the share of interior pixels with a different threshold δ. During inference, we take the average across the three channels of the output image and apply the inverse of the linear transformation used in training to obtain a depth estimate in the range of $[0, 10]$ meters.

Image Classification. As mentioned in Section 2.1, we implement image classification by asking the InstructCV model if a category is visible in the input image $\mathbf{x}$ using the following template prompt: *"Display %color_1% if the image contains %category%, else display %color_2%"*. We evaluate the accuracy of InstructCV for classification by assessing whether $\mathbf{v}(\mathbf{y})$ contains the color block corresponding to the correct category. We do so by generating image pairs based on the Oxford-III Pet dataset for binary classification and augment these with negative pairs, where the category mentioned in the language instruction is not present. We then evaluate a classification score defined as: $\text{Cls-Score}\left(\mathbf{v}(\mathbf{y}), \mathbf{c}\right) = \sum_{i=1}^{n}\sum_{j=1}^{m} |\mathbf{v}_{i,j}(\mathbf{y}) - \mathbf{c}_{i,j}|$, i.e., the Euclidean distance between the pixel-wise colors of the output image $\mathbf{v}(\mathbf{y})$ and target color block $\mathbf{c}$ specified in the task instruction $\mathcal{I}$.

Our pooled multi-modal/multi-task instruction-tuning dataset comprises 180,285 images. We create two versions of the dataset, $\mathcal{D}_{\mathcal{I}}^{\text{FP}}$ & $\mathcal{D}_{\mathcal{I}}^{\text{RP}}$, with fixed and rephrased prompts as described in Section 2.1.

External Datasets. Since the generalist baselines and InstructCV were trained on different datasets[1], we consider additional external datasets that are outside of the training distribution of all baselines. To this end, we consider the following datasets: ImageNet [47] for classification, SUNRGB-D [48] for object detection and VOC [49] for segmentation and monocular depth estimation tasks.

Implementation Details. We train InstructCV for 20 epochs on 8 NVIDIA A100 GPUs over 10 hours. The training involves images with a resolution of 256×256 and incorporates data augmentation including random horizontal flipping and cropping with a batch size of 128. The proposed model is initialized with EMA weights obtained from the Stable Diffusion checkpoint, and trained with a learning rate 10^{-4} without any warm-up stage. Further details can be found in appendix A.1. We refer to the models trained on $\mathcal{D}_{\mathcal{I}}^{\text{FP}}$ and $\mathcal{D}_{\mathcal{I}}^{\text{RP}}$ as InstructCV-FP and InstructCV-RP, respectively.

4.2 RESULTS

Table 1 presents a quantitative comparison between InstructCV and task-specific as well as generalist vision models in both in-distribution and out-of-distribution datasets. (Fig. 4 displays illustrative examples of InstructCV outputs.) For depth estimation, we compare InstructCV with DepthFormer [50], BinsFormer [51] and UviM [12]. For semantic segmentation, our baselines include Mask2Former [53]

[1]Unified-IO has been trained on 90 datasets while Pix2SeqV2 was trained on MS-COCO.

Table 1: **Comparison of InstructCV to task-specific and vision generalist baselines**. We report performance on segmentation, object detection, depth estimation and classification using test samples from the InstructCV instruction tuning dataset as well as external datasets for each task. InstructCV was evaluated using LLM-rephrased instructions for test data. (**Bold** and blue indicate best and second best performing model, respectively.)

	Depth Estimation		Semantic Segmentation		Classification		Object Detection	
	RMSE↓		mIoU↑		Acc↑		mAP@0.5↑	
	NYUv2	†SUNRGB-D	ADE-20K	†VOC	Oxford-Pets	†ImageNet-sub	COCO	†VOC
Task-specific models								
DepthFormer [50]	0.339	0.427						
BinsFormer [51]	0.330	0.433						
SSA [52]			47.150	*				
Mask2Former [53]			**56.100**	*				
ResNet [54]					97.500	*		
ViT [55]					**98.400**	*		
DETR [28]							**60.600**	*
Mask R-CNN [56]							60.500	*
Generalist framework, Task-specific models								
UviM [12]	0.468	*						
Generalist models								
Unified-IO [15]	0.387	0.287	25.713	27.724	96.514	**89.877**		
Pix2SeqV2 [17]							57.400	38.500
Painter [18]	0.288	0.285	49.9	52.5				
InstructCV-FP	**0.275**	**0.268**	52.3	**53.5**	80.4	75.1	49.1	**62.00**
InstructCV-RP	0.297	0.279	47.235	52.125	82.135	74.665	48.500	61.700

† **External datasets**: These dataset were not part of the training dataset for the baselines under consideration.
* Task-specific models are incapable of directly implementing the corresponding task on external datasets if these datasets contain object categories that were not present in the models' training data.

and SSA [52]. For classification, we consider baseline classifiers with ResNet [54] and ViT [55] backbones. Lastly, for object detection, we consider DETR [28] and Mask R-CNN [56]. Task-specific models for semantic segmentation, classification, and object detection have not been assessed on datasets beyond their training distribution. This is due to the fact that these new datasets introduce categories absent in the model's original training, which precludes zero-shot generalization. We consider Unified-IO [15] and Pix2SeqV2 [17] as generalist vision baselines. Note that we only report object detection results for Pix2SeqV2 [17], as this model does not cover the other tasks involved in the development of InstructCV. Additional tasks Pix2SeqV2 is able to perform, such as keypoint detection, have not been integrated into the current version of InstructCV.

Performance comparisons. Overall, Instruct CV performs competitively compared to both generalist and task-specific baselines across all four tasks. For depth estimation within in-distribution data, InstructCV achieves a 10% improvement in RMSE compared to the second best model, BinsFormer [51]. Notably, InstructCV demonstrates strong generalization performance to unseen datasets, surpassing all baselines by a large margin, with the exception of classification tasks. For instance, for the task of depth estimation the task-specific models Binsformer [51] and DepthFormer [50] experience high performance drops of $\approx 31.2\%$ and $\approx 26.0\%$, respectively, while InstructCV's performance improved by 6% resulting in a 34.7% lower RMSE than the best task-specific model. Similarly, for the task of object detection, the performance of the Pix2SeqV2 generalist model drops by 32.2% when evaluated on VOC—InstructCV outperforms this generalist model by a $+23.2$ in mAP@0.5. Among all baselines, only Unified-IO demonstrates comparable generalization properties to unseen datasets. However, InstructCV outperforms Unified-IO on all tasks except for classification. Notably, for semantic segmentation, InstructCV surpasses the performance of Unified-IO by $+24.401$ in mIOU. Classification is the task where InstructCV exhibited its weakest performance compared to baselines.

Table 2: **Open-vocabulary segmentation on the FSS-1000 dataset.**

Model	mIoU ↑
Inpainting [57]	58.5
Generalist Painter [18]	62.3
InstructCV-RP	**69.8**

Generalization to unseen categories. Most existing task-specific and multi-task models are built based on a fixed category pool determined by their training data, and do not exhibit zero-shot capabilities in detecting, segmenting or classifying categories outside of this set [15, 17, 28, 52–

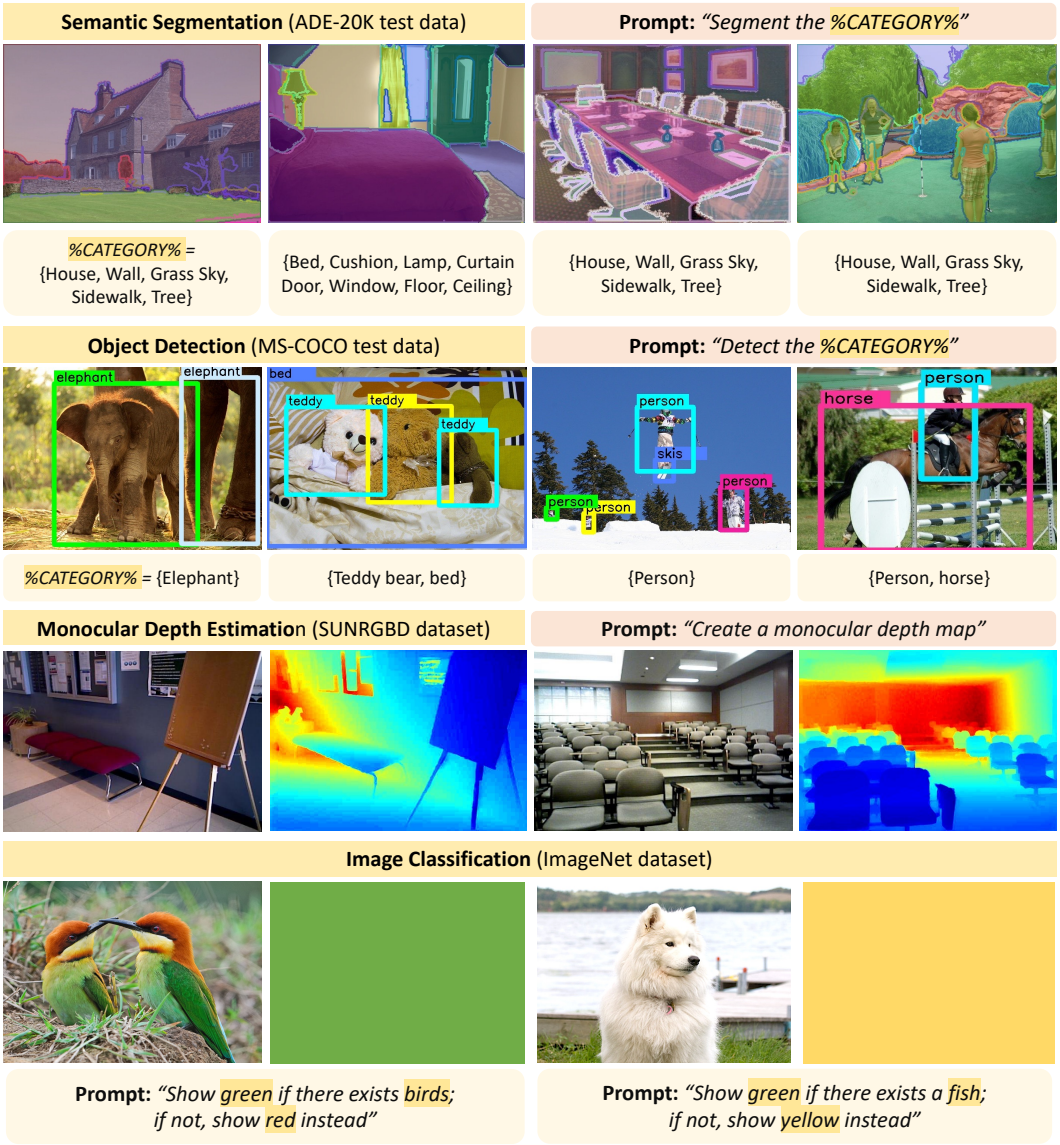

Figure 4: **Samples of InstructCV outputs across all vision tasks.** The segmentation and detection outputs (top two rows) are obtained by applying one prompt for each category and overlaying the results in one output image.

[56]. Because InstructCV leverages semantically meaningful instructions and a pre-trained text-to-image generative model to guide learning, we expect its task-specific capabilities to generalize to new categories. To investigate its generalization capabilities to unseen categories, we evaluate InstructCV on an open-vocabulary segmentation task using the FSS-1000 dataset [58]. This dataset comprises 1000 object classes, many of which have not been previously annotated in other computer vision datasets. Because the baselines in Table 1 do not accommodate many of these categories, we instead compare InstructCV with generalist vision models that exhibit zero-shot capabilities—the visual prompting by Inpainting model in [57] and the Generalist Painter model in [18]. Both methods rely on visual prompting approaches, where the input prompt is a set of pixels and the vision tasks are all represented within a unified inpainting framework. Table 2 presents the results. Overall, InstructCV outperforms Generalist Painter and Inpainting by $+3.5$ and $+7.3$ in mIoU, respectively. This demonstrates the value of repurposing text-to-image models and language-based prompts to improve the generalization capabilities of generalist approaches to computer vision tasks.

Generalization to new user-written instructions. The InstructCV-RP model undergoes training using the dataset $\mathcal{D}_{\mathcal{I}}^{\text{RP}}$, which comprises a variety of instructions, all aimed at conveying a common

Table 3: **Evaluating InstructCV on new user-written instructions.** We evaluate the InstructCV-FP and InstructCV-RP models on the semantic segmentation task using test samples within the ADE20k dataset. At inference time, we apply new user-written instructions (listed below) that have not been used during training in $\mathcal{D}_{\mathcal{I}}^{\text{FP}}$ and $\mathcal{D}_{\mathcal{I}}^{\text{RP}}$, in addition to the template prompt (first row, shaded).

Instruction	**mIOU Score**	
	InstructCV-FP	**InstructCV-RP**
Template prompt: *"Segment the %category%."*	45.503	42.793
"Segment %category%."	36.315	48.739
"Can you help me segment the %category%?"	44.035	45.686
"Highlight the image parts corresponding to the %category%."	19.988	44.758
"Please highlight the image segment containing %category%."	21.952	40.153
"Which image segments contain the %category%?"	39.348	47.972

underlying intent (i.e., describing a specific visual task). Through this diverse training data that encompasses a broad spectrum of phrasings and descriptions of the same task, we expect that InstructCV-RP will be able to extrapolate its learning to novel user-generated instructions at inference time. To test this, we compared the performance of the InstructCV-FP and Instruct-RP variants on the semantic segmentation task. Both models were tested using manually-selected prompts (unseen in training data) on 200 images in the ADE20k test data. The results in Table 3 indicate that the InstructCV-RP model exhibits consistent performance and more robustness to variations in the phrasing of user instructions compared to InstructCV-FP. For example, testing InstructCV-FP with the instruction *"Please highlight the image segment containing %category%."* instead of the template prompt *"Segment %category%."* led to a 51.8% drop in mIOU. Conversely, InstructCV-RP only incurred a performance reduction of 6% with this instruction compared to the template prompt. This suggests that our LLM-based prompt rephrasing approach effectively enhances the ability of InstructCV to generalize to new user-generated prompts that convey descriptions of tasks similar to those seen during training.

Computational costs. Finally, we note that InstructCV was trained in an end-to-end fashion, with only 2,000 finetuning steps. The inference time of InstructCV on a single NVIDIA A100 GPU is 5 seconds (for a 256x256 image). This is a significant improvement over comparable generalist models, such as Unified-IO [15], which was trained from scratch using 1.5 million steps and takes around 40 seconds for inference on a single NVIDIA A100 GPU. Notably, InstructCV not only simplifies the training process but also outperforms Unified-IO in various tasks. These improvements can be attributed to the already impressive capabilities of the underlying text-to-image model. By instruction-tuning a generative model with a relatively small number of steps and a moderately-sized dataset, we are able to steer its functionality with performance that is competitive with bespoke generalist models.

5 Conclusion

In this paper, we introduce a unified language interface for computer vision tasks, dubbed InstructCV, eliminating the need for task-specific design choices and allowing for task execution based on natural language instructions. InstructCV frames various computer vision tasks as text-to-image generation problems. In this setup, textual instructions describe the task, and the resulting image serves as a visual representation of the task output. Following the InstructPix2Pix architecture, we curate a multi-task and multi-modal dataset to instruction-tune a pre-trained text-to-image diffusion model, steering its function from a generative model to an instruction-guided multi-task vision learner. By harnessing semantically meaningful language instructions to drive the learning process, our model demonstrates compelling generalization capabilities across unseen data, categories, and user instructions.

Limitations. The inference speed of our model lags behind specialized task-specific models and falls short of meeting the real-time inference requirements for tasks such as object detection and segmentation. Additionally, the semantic flexibility of InstructCV is constrained by the richness and diversity of our instruction-tuning dataset, which is currently generated by rephrasing a limited set of template prompts. This raises questions for future work: can this learning paradigm accommodate instructions that introduce more nuanced conditions? For example, an instruction might cap the count of objects to be detected. Exploring such ideas might require the integration of strategies such as learning from human feedback, which could enable more versatile generalist models by improving alignment of task outputs with more complex prompts.

6 ACKNOWLEDGEMENTS

The authors would like to thank David Sontag (MIT) for insightful feedback and discussions.

REFERENCES

[1] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[2] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[5] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. *International Conference on Learning Representations*, 5 2021.

[6] Alex Nichol, Prafulla Dhariwal, A. Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. *arXiv (Cornell University)*, 12 2021.

[7] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2015.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[9] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision, 2021.

[10] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *CoRR*, abs/2202.03052, 2022.

[11] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. 2022.

[12] Alexander Kolesnikov, † André, Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. 2023.

[13] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks, 2023.

[14] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks, 2022.

[15] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. 2022.

[16] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*, 2022.

[17] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J. Fleet, and Geoffrey Hinton. A unified sequence interface for vision tasks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[18] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. *arXiv preprint arXiv:2212.02499*, 2022.

[19] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A. Efros. Visual prompting via image inpainting. *arXiv preprint arXiv:2209.00647*, 2022.

[20] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianfeng Gao, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. *arXiv preprint arXiv:2303.08131*, 2023.

[21] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Xiaogang Wang, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. *arXiv preprint arXiv:2112.01522*, 2021.

[22] Hao Li, Jinguo Zhu, Xiaohu Jiang, Xizhou Zhu, Hongsheng Li, Chun Yuan, Xiaohua Wang, Yu Qiao, Xiaogang Wang, Wenhai Wang, and Jifeng Dai. Uni-perceiver v2: A generalist model for large-scale vision and vision-language tasks, 2022.

[23] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. Generalized decoding for pixel, image, and language, 2022.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[25] Charlie Nash, Joao Carreira, Jacob C Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models. *Transactions on Machine Learning Research*, 2023.

[26] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[27] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *Lecture Notes in Computer Science*, 2014.

[29] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. *Computer Vision and Pattern Recognition*, 2017.

[30] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 2019.

[31] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[32] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[34] Maxim Kuznetsov Vladimir Vorobev. A paraphrasing model based on chatgpt paraphrases. 2023.

[35] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[36] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[37] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[38] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022.

[39] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[40] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022.

[41] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[42] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18062–18071, 2022.

[43] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection, 2022.

[44] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. ODISE: Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models. *arXiv preprint arXiv: 2303.04803*, 2023.

[45] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.

[46] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012.

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[48] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[49] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8:5, 2011.

[50] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation, 2022.

[51] Li, Zhenyu, Wang, Xuyang, Liu, Xianming, Jiang, and Junjun. Binsformer: Revisiting adaptive bins for monocular depth estimation. *Cornell University - arXiv*, 2022.

[52] Jiaqi Chen, Zeyu Yang, and Li Zhang. Semantic segment anything. https://github.com/fudan-zvq/Semantic-Segment-Anything, 2023.

[53] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022.

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[55] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly,

Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[56] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[57] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A Efros. Visual prompting via image inpainting. *arXiv preprint arXiv:2209.00647*, 2022.

[58] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2869–2878, 2020.

[59] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.