The Effect of Synthetic Data on the Few-shot Object Detection

[1]B.Srujana, [2]Dr.Ch.Buchi Reddy
[1]PG Student, [2]Associate Professor, [1,2]Dept. of Computer Science Engineering, Ellenki College of Engineering & Technology
E-Mail: [1]bodasrujana6@gmail.com, [2]buchireddy2017@gmail.com

Abstract
Few-shot object detection (FSOD) seeks to enhance the detection capabilities of models for novel categories using minimal training examples. However, the limited availability of these examples often constrains the presentation of FSOD models. Although modern developments in text-to-image production comprise resulted in high-quality synthetic images, their utility for FSOD remains uncertain. This study investigates the potential of using synthetic images produce by advanced text-to-image systems to develop FSOD tasks.
We propose a copy-paste pipeline for leveraging synthetic data. Initially, we employ saliency object detection on the created images to locate as well as crop the major object. The produce object is then indiscriminately pasted onto a base dataset image. Additionally, we explore the impact of varying input texts for the text-to-image generator and the quantity of synthetic images on FSOD performance.
To build a representative synthetic training dataset, we emphasize maximizing image diversity through sample-based and cluster-based techniques. Despite these efforts, high false positive (FP) rates for new categories in FSOD present a significant challenge. To mitigate this, we incorporate CLIP, a zero-shot recognition model, keen on the FSOD pipeline that filters out 90% of FPs by applying a similarity score threshold between the detected object and the predicted category's text.
Comprehensive experimentation accomplished on the PASCAL VOC as well as MS COCO datasets demonstrate the usefulness of our approach, achieving up to a 21.9% performance enhancement evaluated to the few-shot baseline

Keywords: FSOD, object detection, Data generation, Novel dataset, synthetic data

INTRODUCTION

In recent years, object detection has seen tremendous progress, with models achieving impressive performance across various benchmarks [2, 4, 24, 33, 34]. The effectiveness of these models, yet, is largely dependent on having accessibility of a large quantity of structured information, which makes generalizing to fresh categories extremely difficult, particularly in situations where annotated data is hard to come by. But with just a few labeled instances, people can pick up on the capacity to identify or detect an unusual object—a skill that traditional models find difficult to imitate.
Few-shot object detection (FSOD) has emerged as a promising approach that aims to mimic this human-like ability by enabling models to find out to distinguish new grouping with solitary a not many explain instances. In FSOD, a model trained on base grouping by ample data (base images) is expected to generalize and distinguish narrative categories by means of just a few examples (novel images). However, the limited number of training samples often restricts the performance of FSOD models, making it a challenging task.
The rise of generative models, particularly text-to-image generators like DALL.E , Imagen, and Stable Diffusion [35], offers a potential solution by producing high-quality images from textual descriptions. These generators can create diverse outputs, suggesting promising applications in industry for addressing few-shot or long-tailed problems. This research explores how synthetic data generated by these models can be leveraged to enhance FSOD tasks, focusing on the use

of synthetic images from the open-sourced Stable Diffusion model.

Usually, deep learning-based techniques need a large quantity of training data. As a result, applying them to real-world situations involving unique items that are absent from widely used object detection datasets is challenging. For object detection, a lot of photos must be annotated, which is expensive and time-consuming. In other situations, getting a large number of photos may even be difficult, such as in the case of medical applications [1] or the discovery of rare species [2]. Furthermore, people may acquire new concepts with little data even at a young age, in contrast to usual deep-learning-based systems [3, 4, 5]. When presented with unfamiliar objects, kids can identify them even if they have only seen them once or a few times.

Our goal in this survey is to give prospective researchers in this nascent field of study an overview of cutting-edge FSOD methodologies. We begin by defining the FSOD problem. Next, we group the existing methods and point out their advantages and disadvantages. Next, we present widely-used datasets and offer benchmark outcomes. Lastly, we highlight typical evaluation issues and Determine research avenues that show promise for directing future studies.

object identification in a few shots (FSOD). FSOD, as illustrated in Fig. 1, attempts to identify novel objects with few annotated examples after pretraining in the first phase on copious amounts of publically available data. As a result, it lessens the workload associated with extensively annotated material within the intended domain.

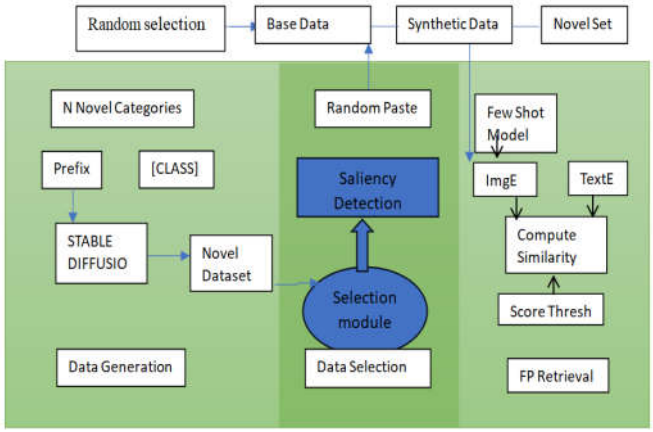


Figure.1. System architecture model

Problem Statement

The central challenge in FSOD is the scarcity of training data for novel categories, which significantly limits the detection performance of existing models. While synthetic data as of text-to-image generators presents an opportunity to overcome this limitation, several critical questions remain unanswered:

1. How can synthetic data be effectively used for FSOD?
2. How can representative samples be selected by large-scale synthetic datasets to maximize their impact on FSOD?

To address these questions, we introduce a new setting for FSOD, termed (K + G)-shot, where K represents the real original instances and G represents the produce original instance. This approach necessitates innovative methods for integrating synthetic data into the FSOD pipeline, ensuring that the synthetic images contribute positively to the model's performance.

Objectives

The objectives by investigation by:

1. Enlarge a methodology for utilizing synthetic images in FSOD tasks. This includes designing a pipeline that can incorporate synthetic data to enhance the detection of novel categories with few real instances.
2. Investigate strategies for identifying representative samples by large-scale synthetic datasets. This involves maximizing the diversity of the training dataset through sample-based as well as cluster-based methods, thereby improving the generalization capability of FSOD models.

LITERATURE REVIEW

Object detection is the collaborative job of locating and categorizing object instances belonging to the categories the detector was trained on. Bounding box coordinates are used to locate regions of interest (RoIs), which are then categorized into a predetermined list of groups. The detector is trained to suppress detections of any other object categories that are not included in the training categories [12]. These other object categories are treated as background. Although these methods yield remarkable outcomes, they necessitate a large number of annotated object instances for each category and generally fall short when implemented in the few-shot regime.

For investigators who are new to this field, we recommend reading thorough studies [13] on this subject.

While there are other FSOD surveys available [6], [7], [8], [9], [10], and [11], they do not include as many articles as ours. The studies in works are more comprehensive and do not concentrate as much on FSOD. They also cover self-supervised, weakly

supervised, and/or zero-shot learning. Because works [6], [7] exclusively address previous research on FSOD, they are a little out of date because they omit some of the methods that are now doing the best on widely used benchmarks. Work [11] is only available to a restricted number of researchers because it is not available in the English language. All things considered, our survey is most similar to [8], since it likewise clarifies a number of fundamental ideas and organizes methods based on these ideas. But thanks to the visual taxonomy the reader may quickly see which approaches adhere to related principles and which concepts seem to work well together. Additionally, we offer more guidance on benchmark results by drawing attention to variations in evaluation processes and classifying methodologies with similar evaluations. Moreover, we cover almost twice as many FSOD studies than [9] did, making our survey far more thorough.

Proposed Model

Few-Shot Object Detection (FSOD) aspires to detect objects from a novel dataset DND_NDN, which consists of only a few samples per class, referred to as the newclasses CNC_NCN. The base dataset DBD_BDB, containing base classes CBC_BCB, has a sufficient amount of images in every class for the model to learn robust feature representation. Importantly, there is no overlap among the base classes CBC_BCB and the new classes CNC_NCN. The FSOD problem is further characterized by the number of instances per class in CNC_NCN, denoted as KKK, where KKK-shot detection refers to the number of available real novel instances per class.

In the context of this study, we introduce a novel setting termed (K+G)(K + G)(K+G)-shot detection, which incorporates both real novel instances KKK and generated novel instances GGG. This setting aims to leverage synthetic data generated via text-to-image models to enhance the FSOD process. The FSOD methodology traditionally involves two key stages: the pre-training stage and the fine-tuning stage.

Pre-training Stage: In this stage, the model is qualified on the base dataset DBD_BDB to learn a robust feature demonstration. This step ensure that the representation can efficiently identify features common across base classes CBC_BCB.

Fine-tuning Stage: After pre-training, the model undergoes fine-tuning on a impartial few-shot dataset, that comprises both base and fresh classes CB∪CNC_B \cup C_NCB∪CN. In our (K+G)(K + G)(K+G)-shot setting, synthetic data is introduced during fine-tuning, attractive the model's capability to simplify across novel classes by leveraging both real and generated instances

Our Pipeline

The proposed pipeline for using synthetic data in FSOD comprises three main steps: Data Generation, Data Selection, and False Positive (FP) Retrieval. These steps are designed to integrate synthetic data effectively into the FSOD process, enhancing the detection performance on novel classes while mitigating issues such as false positives.

Data Generation

Data generation is a critical step in our pipeline, as it involves creating synthetic images that represent the novel classes CNC_NCN. We employ the Stable Diffusion model, a powerful text-to-image age group tool, to produce synthetic images based on carefully designed text prompts. The input text for the generator is formatted in various ways to study its influence on the excellence and diversity of the produced images.

Text Descriptions: We experiment with several representative text descriptions that vary in complexity. For instance, the simplest form uses only the class name as the input text, with no additional descriptors (e.g., "None"). Other variations include prefixes such as "a" or "one" combined with the class name, or more descriptive phrases like "a photo of" or "a picture of," possibly accompanied by adjectives to add specificity.

Generation Process: Using these text descriptions, we generate multiple synthetic images corresponding to each novel class. Most of the produced photos include only one targeted object because the class name is the item in the input text., ensuring that the synthetic data is focused and relevant to the novel class.

Data Selection

Once the synthetic images are generated, the next step is to select the most representative samples for inclusion in the training set. The selection process is designed to maximize the diversity of the particular data, as the excellence of the synthetic images generated by Stable Diffusion is generally high.

Selection Methods: We recommend two mai assortment techniques: sample-based selection and cluster-based selection.

Sample-Based Selection: This method relies on the CLIP model, which computes a correspondence score

among the generated images and the input text or between the generated and real novel images. The goal is to maximize diversity by selecting images that differ significantly from one another, as measured by the cosine similarity between their CLIP features. The selected samples RsR_sRs are those that exhibit the highest diversity while maintaining relevance to the target class.

Cluster-Based Selection: In this method, we use clustering algorithms, such as k-means or spectral clustering, to group the generated images into clusters based on their CLIP features. Each cluster represents a different aspect of the target class. The samples closest to the cluster centroids are selected as representative samples RcR_cRc. Spectral clustering has been found to perform best in our experiments, offering a more nuanced grouping of the generated data.

Saliency Detection and Object Cropping: Once the appropriate samples have been chosen, we use a saliency detection technique to determine which object in every picture is the most prominent. Next, depending on the saliency map, the item is cropped employing the minimum enclosing box. The truncated object is then enlarged and positioned on a backdrop randomly chosen from the underlying dataset.

This process helps integrate the synthetic objects into a more realistic and varied context, enhancing the robustness of the FSOD model.

Hyper-Parameter GGG: The amount of selected synthetic images GGG is a critical hyper-parameter in our (K+G)(K + G)(K+G)-shot setting. This parameter directly impacts the balance between real and synthetic data during training. The optimal value for GGG will be explored and discussed.

False Positive (FP) Retrieval

In FSOD, the false positive (FP) rate, particularly for novel categories, is a significant challenge. Introducing synthetic data alone does not resolve the issue of high FP ratios, as demonstrated in Figure 4 of Section 4.4.4. To address this, we suggest integrating the CLIP model keen on the FSOD framework to filter out false positives.

Detection and Filtering: During detection, the FSOD model FsF_sFs outputs a bounding box bbnbb_nbbn and a confidence score confnconf_nconfn for each detected object. The cropped region RnR_nRn corresponding to bbnbb_nbbn is then passed through CLIP, which computes a resemblance score among the image features and the text features of the novel group tnt_ntn.

Cosine Similarity and Softmax: The similarity score is calculated using the cosine similarity between the image encoder ImgE(Rn)ImgE(R_n)ImgE(Rn) and the text encoder TextE(tn)TextE(t_n)TextE(tn). The input text list tit_iti includes both base and novel categories B∪NB \cup NB∪N or only novel categories NNN. The final score is obtained through a softmax operation over the category list, normalizing the similarity scores.

FP Removal: We define a threshold for the CLIP similarity score, set to 0.1 in our experiments. Any detection results with a score below this threshold are considered false positives and are removed from the final evaluation. By incorporating CLIP, we achieve a significant reduction in the FP ratio, up to 90%, as revealed in Fig 4.

Methodology

Datasets and assessment Protocols

Present paper we present a comprehensive analysis of the datasets and assessment protocols employed in our experiments. The main datasets used for evaluation include the PASCAL VOC as well as MS COCO, which are the benchmarks for various object detection tasks, including few-shot object detection (FSOD).

PASCAL VOC

The PASCAL VOC dataset is widely used in the object detection community. It contains 20 object categories, and we follow the common practice by splitting these categories into base and new classes for the FSOD tasks. Specifically, the 2007 test set is utilized for evaluation, whereas the 2007 as well as 2012 train/validation sets are combined for training purposes. According to the protocols laid out by prior works, we focus on three different splits for the novel and base categories, referred to as $split\ 1, split\ 2, and\ split\ 3$. every split consists of 15 base category by sufficient data and 5 new categories by limited annotated instances ($K = 1, 3, 5, 10$). The performance of novel categories is measured using mean average precision (mAP) at a 0.5 $Intersection - over - Union\ (IoU)\ threshold$.

MS COCO

The MS COCO dataset is added complex, containing 80 object categories. For FSOD, 20 grouping are designated as new, while the remaining 60 grouping serve as base classes. The recognition presentation is

evaluated using COCO-style average precision (AP) and $AP75$ metrics across various $K-shot\ settings$ $(K = 1, 3, 5, 10, 30)$ for the novel categories. This rigorous evaluation protocol ensures a robust comparison across different FSOD methods.

Image noise reduction technique

Certain diffusion models have been more and more well-liked in the last few years, particularly because of their capacity to provide image data with cutting-edge results. Diffusion models, yet, might be operationally costly and memory-intensive to use.

Using the other hand, by conducting the diffusion process to a smaller dimensional latent space, latent diffusion lowers intricacy and memory utilization. The model has been taught to provide compressed illustrations of pictures in latent diffusion.

Latent diffusion consists of three primary parts.

1. VAE, or autoencoder.
2. The United Network.
3. A text encryptor

Auto encoder (VAE)

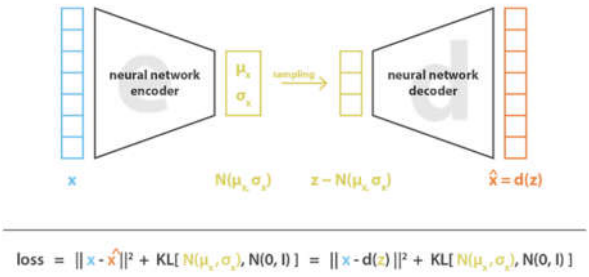
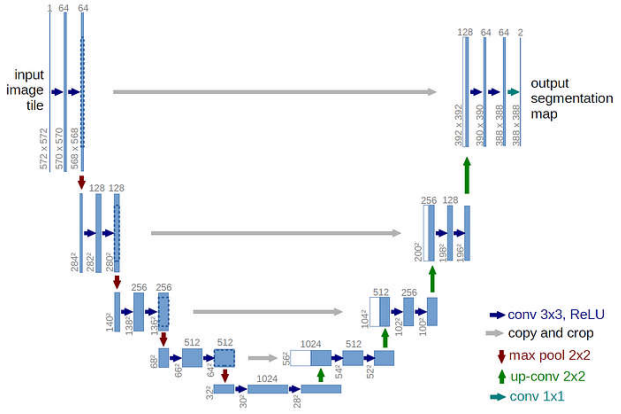


Figure.11.Architecture of the Variation Auto encoder an encoder as well as a decoder are both parts of the Variation Auto encoder (VAE) paradigm. The decoder turns a latent representation within an image whereas the encoder uses the picture to create low dimensionality latent representations that is utilized as an input for the U-Net models.

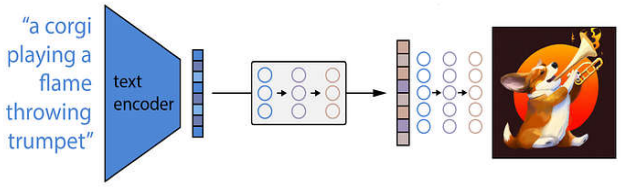
U-Net



The U-Net architecture

One popular convolutional neural network for image segmentation tasks is the U-Net. It additionally includes two ResNet block encoders and a decoder. An picture is compressed by the encoder becoming a lower resolution image, that is then returned to the original higher density—which is meant to be less noisy—by the decoder.

Text Encoder



How is text encoders operated?

The text input prompts must be converted by the text encoder hooked on an area of embedding that the U-Net can comprehend. An encoder that maps a series of input tokens to a series of latent text embeddings is often a straightforward transformer-based encoder.

Implementation Details

This section outlines the technical implementation of our methods, including the baselines and models used.

Baselines

We adopt DeFRCN and MFDC as our baseline models, mutually depending on the Faster R-CNN architecture with ResNet-101 as the backbone. These baselines are representative of state-of-the-art FSOD techniques, providing a solid foundation for assessment. The training strategies for our methods align with these baselines, ensuring a fair comparison.

Use of Synthetic Data

A significant innovation in our approach is the inclusion of artificial data in the few-shot training set. Specifically, we generate additional novel instances using a text-to-image creator and incorporate these into

the few-shot set. For each K-shot setting, we add $G = 20$ artificial images per class, resulting in a (K+G)-shot setup. This additional data helps bridge the gap between the base and novel classes, improving detection performance.

CLIP Encoder

We leverage the ViT-B/32 architecture from CLIP as the encoder in our pipeline. CLIP's ability to align images and text semantically is crucial for filtering and refining the synthetic data, ensuring that only high-quality instances contribute to the training process.

Assessments by State-of-the-Art techniques

This section presents a detailed assessment of our approach by existing state-of-the-art FSOD methods, demonstrating the efficacy of our innovations.

PASCAL VOC Results

We estimate our methods on the $PASCAL\ VOC$ dataset across the three splits mentioned earlier. Our results, presented in Table 2, show a consistent improvement over the baselines in all few-shot settings, with the most significant performance gain reaching up to 21.9%. This improvement underscores the effectiveness of our synthetic data generation and selection process.

MS COCO Results

Similarly, our methods achieve $state-of-the-art$ performance on the $MS\ COCO$ dataset, as shown in Table 3. While the performance gain is more pronounced in the lower shot settings, it diminishes as the number of shots increases. This trend suggests that the benefit of synthetic data is most significant when the available annotated data is scarce.

Ablation Study

To further analyze the contributions of different apparatus in our pipeline, we conduct a series of ablation studies on $PASCAL\ VOC$ split 1 using $DeFRCN$ as the baseline.

Components of the Pipeline

Our pipeline consists of four key steps: Add, Crop, Select, and CLIP. Each step is crucial for optimizing the use of synthetic data in FSOD.

Add: This step involves copying the unique synthetic images and pasting them onto a backdrop. This simple augmentation provides a baseline for comparing the impact of more sophisticated techniques.

Crop: We use saliency detection to identify and crop the main target in the synthetic images before pasting. This step is critical, as it considerably get better the exactness of the bounding boxes, resultant in a $4\%\ to\ 10\%$ enhancement in $AP50\ on\ PASCAL\ VOC$.

Select: We implement a selection strategy to identify the top 20 delegate samples by the synthetic dataset. This step refines the training set, improving precision by up to 2% on $PASCAL\ VOC$.

CLIP: The CLIP model is used to filter false positives (FPs) by manipulative the similarity among the input image and text images of the categories. This step reduces FPs by up to 90%, further enhancing performance.

Data Generation and Selection Factors

We investigate three critical factors in synthetic data generation and assortment: input text format, the numeral of synthetic images, and the pasting method.

Input Text Format: Different text descriptions for generating synthetic images impact the performance. We find that the "a5" and "one5" formats, which provide concise and descriptive prompts, yield the best results. We standardize the generation process by using 20 synthetic images per class in all experiments.

Pasting Methods: We compare three methods for pasting synthetic instances onto backgrounds: box, saliency map, and segmentation map. The box method, which uses the least amount enclosing rectangle of the saliency map, consistently delivers the best performance.

Selection Module

We explore various selection strategies to desire 20 samples from the 200 produce images per class. Strategies focusing solely on quality, such as CLIP max, perform poorly due to reduced data diversity. Conversely, methods that emphasize diversity, such as spectral clustering, outperform others, highlighting the importance of maintaining a diverse training set.

CLIP Utilization

We analyze the effectiveness of the CLIP model in reducing false positives. By using CLIP to filter predictions, we achieve a significant reduction in FPs, improving overall detection accuracy. Additionally, we compare two input category lists for the text encoder of CLIP: one containing together base and narrative categories, and the other containing only novel categories. The former approach, which utilizes the full category list, yields optimal performance, particularly when the CLIP threshold is set to 0.1.

Additional Analysis

Finally, we explore the upper bounds of presentation when by means of synthetic data. As the amount of synthetic data increases, the performance gains diminish, particularly when using copy-paste methods. We also compare the performance of different text-to-image generators, like GLIDE and Stable Diffusion, demonstrating that the quality of the generator significantly impacts FSOD performance.

The experiments conducted in this chapter offer strong verification of the efficiency of our approach in attractive FSOD performance through the use of synthetic data and advanced selection techniques. Our methods consistently outperform state-of-the-art FSOD techniques, particularly in low-shot settings, underscoring the assessment of artificial data in overcoming the challenges of limited annotated data. The insights gained from our ablation studies further refine our understanding of the critical factors that contribute to the success of FSOD models.

Object Detection

Although it has come a long way, computer vision still struggles to match human perception's accuracy. The topic of this paper is computer vision. We'll be starting starting scratch here. For novices, differentiating among comparable computer vision tasks may prove difficult. When machine learning is used to identify objects in an image, humans can do so with ease. Despite minimal awareness, the human visual system can quickly and accurately carry out complicated tasks including recognizing multiple objects and distinguishing obstacles. Large volumes of data, quicker GPUs, and improved algorithms have made it possible for us to train computers to accurately identify and categorize several items in an image.

Using this type of recognition and the practice of localization object detection can precisely label items, count the number of objects in a scene, and monitor each object's exact location.

Object detection: what is it?

In computer vision, object detection is the process of locating things in pictures or movies. In order to produce useful results, such algorithms often depend on machine learning as well as deep learning techniques. Presently, let's use the illustration following to assist us compress this statement a little.

Figure.2.

Therefore, we must really identify a dog in the image rather than categorizing the type of dog that exists in these photographs. In other words, I need to determine where the dog is located in the picture. Is it at the bottom left or in the center? And so forth. The next thought that crosses people's minds is, "How can we do that?" Now let's get going.

Therefore, we can draw a box surrounding the dog in the picture and give it specific coordinates for both x and y.

Figure.3.

For the time being, assume that these boxes' dimensions can be used to indicate where an item is in the image. Hence, the box that encloses the object in the picture is referred to as a bounding box. This turns into an image localization challenge now, since we have to figure out wherever the object is located in a group of

photographs.

Take notice that there is just one class here. What occurs if there are several classes? For instance:

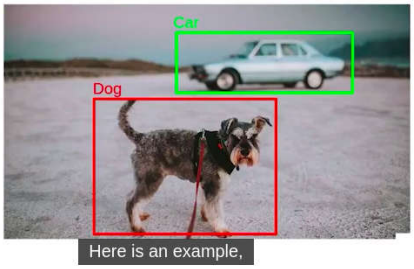


Figure.4.

We must find the objects in this picture, but keep in mind that not all of them are dogs. We have a car and a dog here. Therefore, in addition to finding the objects in the picture, we also need to identify if the object we find is a dog or a car. This then turns into an issue with object detection.

When dealing with object detection issues, we must both categorize the things in the picture and determine their locations. However, there was only one goal in the picture classification challenge, and that was to categorize the objects in the image.

We will talk about the intersection over the union (IoU), and these is a really fascinating idea, in this part. And we're going to employ this to figure out what the target variable is for each of the customized patches we've made.

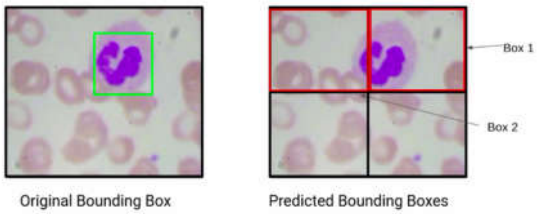


Figure.5. Original bounding box vs predicted bounding boxes

Thus, think about the following situation. Two boxes with borders, box1 and box2, are shown here. If I were to ask you tonight whichever of the following boxes is more correct, box 1 would be the clear choice.

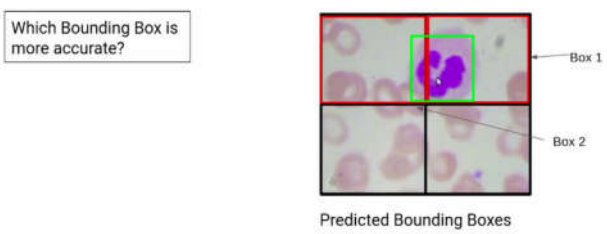


Figure.6. Predicted bounding boxes

Why? due to the has successfully identified the WBC and contains a substantial portion of the WBC.

However, exactly can we quantitatively determine this? Thus, contrast the projected and reality bounding boxes. Determining the extent of overlap between the actual as well as anticipated bounding boxes will allow us to determine whether bounding box is an accurate prediction.

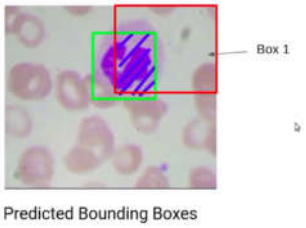


Figure.7. Predicted bounding boxes

Therefore, a better forecast is the bounding box through a higher overlap over the real bounding box. For the initial box, box1, the overlap has been referred to as the area of collision. We can estimate that the intersection's area makes up roughly 70% of the boundaries.

On the other hand, if you look at box 2, the area wherein the second box of boundaries and the real bounding box overlap is almost 20%.

Results & Analysis

A text-to-image strategy called Stable Diffusion was trained using 512x512 photos from a portion of the LAION-5B dataset.

This notebook aims to show how simple it is to build text-to-image creation employing the state-of-the-art pre-trained model of diffusion for images, audio, and 3D designs included in the ▪ Diffusers library. But first, we must define stable diffusion precisely prior to we can begin coding.

Artificial intelligence models of the diffusion kind are trained to extract an instance of interest by denoising an object, like an image. Up as long as a sample is obtained, an algorithm gets taught to gradually denoise the image. Following is a screenshot of this procedure.

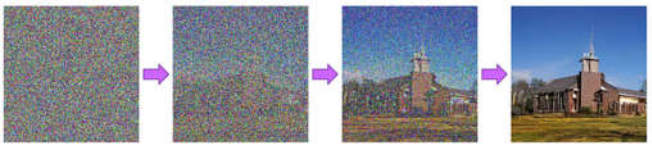


Figure.10. Denoising images

Pipeline for Stable Diffusion with Diffusers

The Diffusers library developed the StableDiffusionPipeline, a pipeline that requires only a few lines of Python code to produce images from text. You may view the many versions and checkpoints that are available by going to the library documentation's Text-to-Image Generation page.

We will be using Stable Diffusion version 1.4 $(CompVis/stable-diffusion-v1-4)$ for this notebook. In order to lower the amount of memory consumed, we are additionally loading the fp16 weights using torch_dtype = torch.float16.

Installing the diffusers into the library will be the first step in our work

$\#Installing\ diffusers\ library$

$!pip\ install\ diffusers$

Now, we may import all necessary libraries.

$\#Library\ imports$

$\#Importing\ pytorch\ library, for\ building\ and\ training\ neural\ networks\ import\ torch$

$\#Importing\ StableDiffusionpipeline\ to\ use\ pre\ trained\ stable\ diffusion\ models\ from\ diffusers\ import\ stablediffusionpipeline$

$\#Image\ is\ a\ class\ for\ the\ PIL\ module\ to\ visualize\ images\ in$

Let's establish a pipeline instance.

$The\ .from\_pretrained$(compVis/stable-diffusion-v1-4")

To create images from text, the diffusion model will be initialized using pre-trained weights and settings. Additionally, a trained VAE, U-Net, and text encoder will be used.

$The\ torch_{dtype} = torch.float16\ sets$

To aid in inference performance, the model's data type was set to float16, a lower-precision floating-point format.

$\#creating\ pipeline$

$pipeline$

$= stablediffusionpipeline.from_{pretrained}$

(compVis/stable-diffusion-v1-4, $torch_{dtype}$

$= torch.float16)$

We can now define an algorithm that will use stable diffusion to create and show an image grid.

$\#define\ function\ for\ the\ creation\ of\ a\ grid\ of\ images$ def $image_{grid(imgs,rows,cols)}$

$assert\ len(imgs) == rows * cols$

$w, h = imgs[0].size$

$grid = image.new\left('RGB', size\right.$

$\left. = (cols * w, rows * w)\right)$

$grid_w, grid_h = grid.size$

$for\ i, img\ in\ enumerate(imgs): grid.paste\big(img, box = (i\%cols * w, i * h)\big)$

$return\ grid$

Next, we transfer our pipeline to the GPU using PyTorch's to function, which accelerates neural network training and inference.

$\#moving\ pipeline\ to\ GPU$

$pipeline = pipeline.to('cuda')$

Finally, we can create graphics from text using Stable Diffusion!

To specify how many photos will be generated, the code below uses n_images. Prompt, on the other hand, is the words that will be utilized to create the desired pictures.

$n\_images = 6$

$Prompt = [Sunset\ on\ a\ beach] * n\_images$

$images = pipeline(prompt).images$

$grid = image\_grid(images, rows = 2, cols = 3)$

$grid$

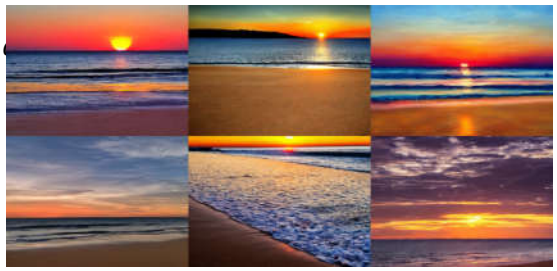


Figure.11 A beach at sunset

$n\_images = 6$

$Prompt = \left[portrait\ of\ Napoleon\ bonaparte'\right] * n\_images$

$images = pipeline(prompt).images$

$grid = image\_grid(images, rows = 2, cols = 3)$

$grid$

Figure.12. Napoleon Bonaparte's portrait

$n\_images = 6$

$prompt = [$'Skyline of a cyberpunk megalopolis'] * n_images

$images = pipeline(prompt).images$

$grid = image\_grid(images, rows = 2, cols = 3)$

$grid$

Figure.13. A cyberpunk megalopolis' skyline

$n\_images = 6$
$prompt = [$'A woman painted in the manner of Van Gogh'$] *$ n_images
$images = pipeline(prompt).images$
$grid = image\_grid(images, rows = 2, cols = 3)$
$grid$

Figure.14. A woman painted in the manner of Van Gogh

$n\_images = 6$
$prompt = [$'Picture of an astronaut in space'$]$
$*$ n_images
$images = pipeline(prompt).images$
$grid = image\_grid(images, rows = 2, cols = 3)$
$grid$

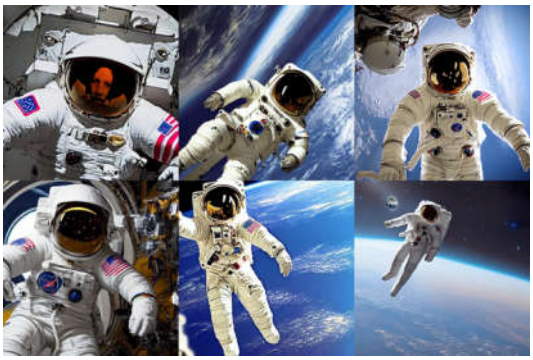


Figure.15. An image of a space astronaut

$n\_images = 6$
$prompt = [$'Renaissance sculpture of a marble bus'$]$
$*$ n_images
$images = pipeline(prompt).images$
$grid = image\_grid(images, rows = 2, cols = 3)$
$grid$

Figure.16. A marble bust sculpture from the Renaissance

Conclusion

In the present study, we make use of externally generated synthetic new data for FSOD through a text-to-image generator. We concentrate on using the synthetic novel data, i.e., the data selection approach for finding the representative synthetic data, the amount of employed synthetic photos, the copy-paste pipeline, and the input text architecture for Stable Diffusion. In addition, we suggest including CLIP with the FSOD paradigm to address the issue of a high FP ratio in the framework. The efficiency of our technique is validated through extensive tests conducted on the few-shot recognition datasets, namely PASCAL VOC as well as MS COCO.

References

1. Katzmann, O. Taubmann, S. Ahmad, A. Mühlberg, M. Sühling, and H.-M. Groß, "Explaining clinical decision support systems in medical imaging using cycle-consistent activation maximization," Neurocomputing, vol. 458, pp. 141–156, Oct. 2021.
2. L. Mannocci et al., "Leveraging social media and deep learning to detect rare megafauna in video surveys," Conservation Biol., vol. 36, no. 1, Feb. 2022, Art. no. e13798.
3. L. B. Smith, S. S. Jones, B. Landau, L. Gershkoff-Stowe, and L. Samuelson, "Object name learning provides on-the-job training for attention," Psychol. Sci., vol. 13, no. 1, pp. 13–19, Jan. 2002.
4. L. K. Samuelson and L. B. Smith, "They call it like they see it: Spontaneous naming and attention to

shape," Develop. Sci., vol. 8, no. 2, pp. 182–198, Mar. 2005.
5. L. A. Schmidt, "Meaning and compositionality as statistical induction of categories and constraints," Ph.D. dissertation, MIT, Cambridge, MA, USA, 2009
6. S. Antonelli et al., "Few-shot object detection: A survey," ACM Comput. Surv., vol. 54, no. 11, pp. 1–37, 2022.
7. L. Jiaxu et al., "A comparative review of recent few-shot object detection algorithms," 2021, arXiv:2111.00201.
8. T. Liu, L. Zhang, Y. Wang, J. Guan, Y. Fu, and S. Zhou, "An empirical study and comparison of recent few-shot object detection algorithms," 2022, arXiv:2203.14205.
9. Q. Huang, H. Zhang, M. Xue, J. Song, and M. Song, "A survey of deep learning for low-shot object detection," 2021, arXiv:2112.02814.
10. G. Huang, I. Laradji, D. Vazquez, S. Lacoste-Julien, and P. Rodriguez, "A survey of self-supervised and few-shot object detection," 2021, arXiv:2110.14711.
11. C. Liu et al., "A survey of few-shot object detection," J. Frontiers Comput. Sci. Technol., vol. 2022, pp. 1–15, Feb. 2022.
12. Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 11, pp. 3212–3232, Nov. 2019.
13. L. Liu et al., "Deep learning for generic object detection: A survey," Int. J. Comput. Vis., vol. 128, no. 2, pp. 261–318, Feb. 2020