HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 4 (2023)

Towards Sustainable AI: Mitigating Carbon Impact Through Compact Models

Eddie Zhang[1#], Jixiu Chang[1#] and Ashley Yang[1#]

[1]Troy High School
[#]All authors made equal contributions

ABSTRACT

As AI technology continues to advance rapidly, it is essential to address the environmental concerns associated with the increasing carbon emissions and their contribution to global warming. The expanding AI industry requires significant computing power, making it a potential major contributor to carbon emissions in the future. Unfortunately, our current understanding of AI models is very limited. We conducted a comprehensive analysis involving 12 distinct AI models, encompassing object detection, translation, and text-to-image generation tasks. Our findings revealed that smaller AI models can achieve equal or even better results compared to larger models while offering a significant reduction of carbon emissions. This highlights the potential for environmental savings by prioritizing smaller models. These findings underscore the importance of considering the environmental impact of AI models and encourage the adoption of strategies such as using smaller models and optimizing workload schedules to reduce carbon emissions. By prioritizing sustainability in AI development and deployment, we can work towards a greener and more sustainable future.

I. Introduction

The carbon emissions generated by AI models are a pressing concern as the demand for AI technologies continues to grow rapidly. While AI offers various practical applications in tasks such as object detection, translation, and content generation, its environmental impact is often overlooked.

The widespread adoption of AI, as exemplified by ChatGPT's immense user base, indicates the exponential growth of the industry in the coming years. However, this growth poses a challenge due to the substantial computing power required to train and run AI models. This high energy consumption, particularly when powered by non-renewable sources, results in significant carbon dioxide emissions.

Unfortunately, there is a widespread lack of awareness among the general public regarding the significant variations in carbon emissions resulting from different activities. Furthermore, the field of Green AI, which focuses on mitigating the environmental impact of artificial intelligence, is still in its early stages of development. As a result, due to the absence of adequate metrics and tools, it is commonly assumed that larger AI models consistently outperform smaller ones. However, this paper investigates the effectiveness of employing smaller AI models that generate fewer carbon emissions while still attaining comparable, if not superior, performance when compared to their larger counterparts.

By prioritizing smaller models, we can limit the scaling of carbon emissions associated with larger models. Through our comprehensive analysis of 12 AI models, which are among the most up-to-date and most downloaded models available on Hugging Face (Hugging Face models, 2017), in the domains of language translation-Facebook wmt19-en-de and wmt19-de-en(Ng et al., 2019), AllenAI wmt19-de-en-6-6-base and wmt16-en-de-12-1(Pappas et al., 2019), and Google bert2bert_L-24_wmt_de_en(Lewis et al., 2020), object detection-Deformable DETR(Zhu et al., 2020), OwlVit-base-patch14(Minderer et al., 2022), OwlVit-base-patch32(Minderer et al., 2022), DETR-Resnet(Carion et al., 2020), and image generation. Lastly, for text-to-image generation, we analyzed the default stable

diffusion v2-1(Stable diffusion 2.1, 2023) and 2 other models of different sizes, Bloodorange mix(Abyssorangemix2 - sfw/soft nsfw, 2023) and Pastel mix (Pastel Mix Stylized Anime Model, 2023).

These findings highlight the importance of considering the trade-off between model size and performance, as well as the environmental impact when developing and deploying AI technologies. By adopting smaller models, we can work towards minimizing the carbon footprint of AI and creating a more sustainable future.

The rest of the paper is organized as follows. Section II will discuss the contributions of other papers toward the field of Green AI. Section III will discuss the details of the AI models used in this experiment. Section IV will discuss the findings of our experiment. Section V will discuss the conclusion reached, the limitations our research faced and our plans for future research.

II. Related Work

The issue of carbon emissions in AI has received significant attention from researchers in recent years. Several papers have addressed this concern by exploring various aspects of AI's carbon footprint and proposing potential solutions.

Dhar (Dhar, 2020) highlights the dual nature of AI, which can contribute to both smart grid designs and low-carbon technologies while also generating carbon emissions when used extensively. This highlights the need to find ways to reduce AI's carbon footprint. Schwartz (Schwartz et al., 2020) advocates for a "Green AI" framework that aims to reduce resource consumption while maintaining acceptable results. He calls for further research in this area to address the substantial carbon footprint of AI technologies. Yigitcanlar (Yigitcanlar et al., 2021) emphasizes the importance of collaboration between different stakeholders, including governments, industry, academia, and civil society, to reduce emissions from AI models deployed in large-scale urban environments. This collaborative approach can lead to more effective strategies for achieving the goals of Green AI.

Mehlin (Mehlin et al., 2023) discusses strategies to decrease carbon emissions throughout an AI system's lifecycle, including IT infrastructure, data management, modeling, training, and deployment. This comprehensive approach recognizes the multiple stages where carbon reduction can be achieved. Bergstra (Bergstra et al., 2011) and his team propose a novel algorithm for hyper-parameter optimization, based on sequential model-based optimization (SMBO), which outperforms existing algorithms on benchmark datasets. This optimization technique can contribute to more efficient AI models and subsequently reduce carbon emissions. Dettmers (Dettmers and Zettlemoyer, 2019) presents a method for training sparse neural networks using a regularization technique called "sparse group lasso." This approach allows for faster training without the need for pre-training, contributing to energy-efficient AI models.

Rolnick (Rolnick et al., 2022) suggests leveraging AI in energy optimization applications to mitigate environmental impacts. However, challenges such as data availability and privacy need to be addressed when implementing AI in these contexts. Nishant (Nishant et al., 2020) raises concerns about AI research, including the reliance on historical data, uncertain human responses to AI interventions, increased cybersecurity risks, and challenges in measuring the effects of intervention strategies. These challenges need to be addressed to ensure responsible and sustainable AI development.

While these works focus on improving AI efficiency and reducing its carbon footprint through technical advancements, our work emphasizes simpler solutions. We propose using smaller AI models as an effective and easily implementable strategy to reduce overall carbon emissions in AI systems.

III. AI Models

Object Detection Models

Here we will discuss the 12 AI models selected for this experiment, more specifically the general overview of how they work. These models were among the most recent and most downloaded models available to us on Hugging Face (Hugging Face models, 2017).

1. *Deformable Detection Transformer Model (DETR) - includes Deformable DETR and DETR-Resnet:* The DETR model is a transformer with an encoder-decoder architecture and a convolutional backbone. It incorporates two heads on the decoder outputs to carry out object detection. One head consists of a linear layer that predicts class labels, while the other head comprises a multi-layer perceptron (MLP) that predicts bounding boxes. To detect objects in an image, the model utilizes object queries, which are designed to look for specific objects in the image. The number of object queries for COCO is fixed at 100.(Zhu et al., 2020)(Carion et al., 2020) These two relatively small models, at around 161 MB for Deformable-DETR, and around 168 MB for DETR-Resnet-50, were chosen to test the effectiveness of smaller AI models compared to larger ones. They would also be compared against each other to test for effectiveness and consistency differences with models using the same foundation but different training.
2. *OwlVit - includes OwlVit-base-patch14 and OwlVit-base-patch32:* Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby introduced OWL-ViT (short for Vision Transformer for Open-World Localization) in their paper "Simple Open-Vocabulary Object Detection with Vision Transformers". OWL-ViT is a zero-shot model for text-conditioned object detection that allows users to query an image with one or multiple text inputs. The model employs CLIP as its multi-modal backbone, using a ViT-like Transformer to obtain visual features and a causal language model to obtain text features. (Minderer et al., 2022) These larger models, at around 1.74 GB for patch14 and around 615 MB for patch32, were selected to test if larger models generated better results. Like the DETR models, these two would also be compared against each other to effectiveness and consistency differences with models using the same foundation.

Translation Models

1. *Facebook FAIR's WMT19 News translation submission - German to English and English to German:* The WMT19 is the 2019 Workshop on Machine Translation, which challenges participants to translate certain inputs into different languages. As a result, both were trained using the WMT19 dataset. The Facebook FAIR submission includes a ported German-to-English and English-to-German translation model. Both models are approximately the same size, around 1.08 GB. We classify both models as medium-sized models- between 1 and 2 GB. The German-to-English AI model has a BLEU score of 40.8 and the English-to-German AI model has a score of 42.7.(Ng et al., 2019)
2. *AllenAI's WMT News translation submissions - WMT19 German to English and WMT16 English to German:* Similar to the Facebook FAIR model, the AllenAI model was also made for the WMT, and trained using the WMT19 and WMT16 dataset. We use a ported version of the 2019 German-to-English AI and the 2016 English-to-German AI. Both models are less than 1 GB in size, classifying them as smaller models. The WMT16 model has a BLEU score of 25.75 and the WMT19 model has a BLEU score of 38.37.(Pappas et al., 2019)
3. *Bert2Bert L-24 WMT EncoderDecoder Model - German to English:* The model is a BERT-initialized Seq2Seq model trained for text translation using the WMT14 dataset. The model is 3.09 GB, which classifies

it as a larger model relative to the other translation models we used. The model has a BLEU score of 39.3, which is smaller than the scores of the medium-sized models. (Lewis et al., 2020)

Stable Diffusion - Text to Image and Image to Image Models

Just as the name insists, Stable Diffusion is an image generation AI model that creates images according to the input of the user, using the numerical method stable diffusion. To put it shortly, this is a process that involves using a series of steps to create an image by gradually refining it from an initial noisy image.

The following are the checkpoints under the main model of Stable Diffusion that we deployed.

1. *Stable Diffusion v2-1:* The most basic checkpoints. Has a model size of 5.09GB. Suited for creating all types of images.
2. *Bloodorange mix:* Model suited for creating more "Anime" styled art, has a size of 2.08GB.
3. *Pastel mix:* Model which creates aesthetically pleasing "pastel-like" styled art, has a size of 5.84GB.

The model will start with some input prompt and a random noise image. The model then undergoes a series of steps, named diffusion timesteps, where it denoises the initial image so that it is guided toward an image that matches the description. The model also will reverse the process, having the conditioning vector guide and generate a coherent image that corresponds to the text prompt. Finally, the model samples from the distribution to generate the final image. This last step is called sampling.

We also investigated the 4 main methods of sampling for Stable Diffusion (Stable diffusion samplers, 2023). They are Euler-a, LMS, DDIM, and DPM++ (we will be using 2M Karras). These, by taking different steps of sampling, will alter the time needed for image generation quite a bit.

k-LMS: Using a sequence of small random steps that align with the distribution's gradient to construct the final image. The step size is adjusted based on the curvature of the distribution.

DDIM: An extension of the LMS method. By eliminating variance and bounding the distribution, it refines the outcome and provides more precise sampling compared to LMS. Achieving this entails incorporating additional details into the distribution.

k-Euler: Similar to DDIM, it demonstrates remarkable speed and delivers satisfying results. However, it noticeably alters the style of the image generated compared to DDIM.

DPM++ 2M: Arguably the best performing one among the four sampling methods when it comes to generating precision. it operates at a slower pace but produces consistent and visually captivating images. It excels when working with intricate prompts that carry a low error probability.

IV. Experimental Results

In this section, we will present our experimental results. We will show the different experiments and results, via tables and graphs, for each of the three categories of object detection, translation, and text-to-image generation. It is important to note that all three experiments relied on CodeCarbon (CodeCarbon, 2021) to record carbon emissions.

Object Detection

For object detection, we employed four AI models: Deformable-DETR, OwlVit-base-patch14, OwlVit-base-patch32, and DETR-Resnet. To visualize the results, we utilized PyTorch code(Mukulsomukesh, 2022) to draw bounding boxes based on the coordinates generated by the AI models.

It's important to note that a bounding box represents a rectangular region surrounding an object, specifying its position. Additionally, a label, which is a classifying word or phrase, is assigned to each object and displayed in the top-left corner of the corresponding bounding box.

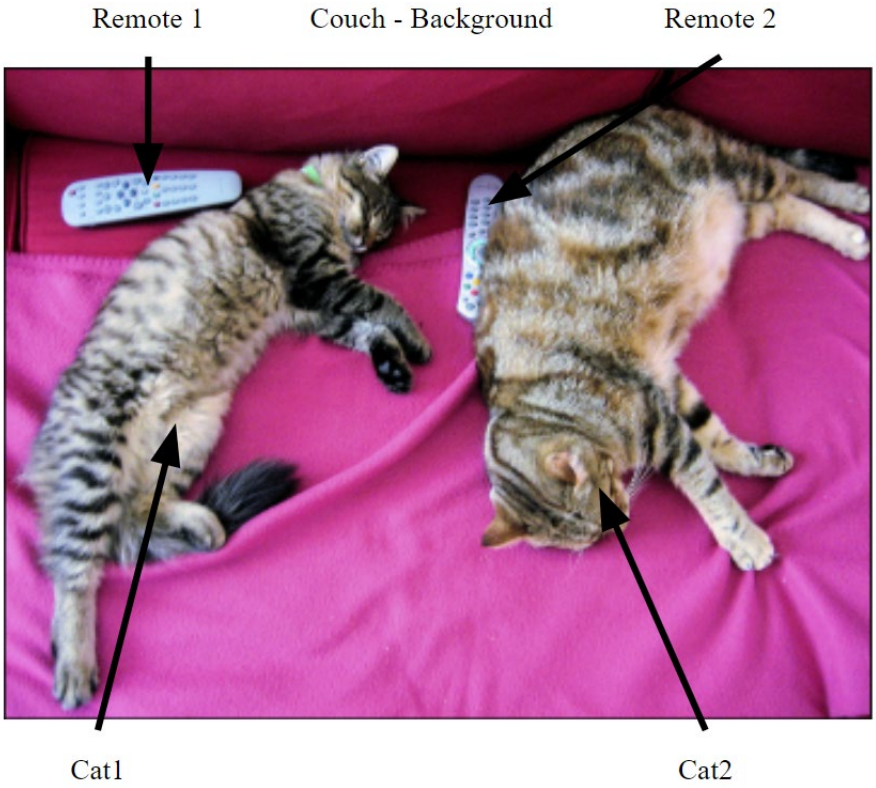
For the four object detection models, we selected five test images shown below and noted down the bounding box coordinates generated by each AI model and inputted them into the Pytorch code (Mukulsomukesh, 2022) to visualize the results.

To evaluate the performance of the four object detection models, we selected a set of five test images. The resulting bounding box coordinates generated by each AI model were fed into the PyTorch code to visualize the detection results. Subsequently, we conducted 15 test runs for each AI model and the results will be presented in the following writing.

In our analysis, we calculated the average carbon emissions for each AI model across all five images, as well as the overall average carbon emissions by averaging the averages of each image. The conclusion will be based on several factors, including the number of objects detected by each AI, the tightness of the bounding boxes, and the carbon emissions produced.

Below, you will find the results indicating the resulting bounding boxes and the area of each bounding box for every image and AI model, along with the corresponding carbon emissions. It is important to note that the y-axis of the carbon emissions figures are in units of ee-8 kg. Additionally, the area of each bounding box is calculated using

Equation 1: *(xmax - xmin) * (ymax - ymin) (1)*



Cats

In the following tables, Cat 1 refers to the cat on the left, Cat 2 refers to the cat on the right, Remote 1 refers to the remote on the top left, Remote 2 refers to the remote on the top middle, and the couch refers to the background.

Table II and Table III show the results for each image processed, for each AI. The first column specifies the AI model used (if a model name is not present, it means that AI did not detect anything). The second column specifies the object detected; these will be identified on the original image for reference. The next four columns specify the coordinates of the bottom left (xmin, ymin) and top right (xmax, ymax) corners of the bounding box. The final column specifies the area of each bounding box as calculated with the formula mentioned previously.

Table 1. Average Carbon Emissions for Cats Image

AI Model	**Emissions(kg)**
Deformable	2.307e-8
OwlVit-base-patch14	2.654e-8
OwlVit-base-patch32	2.688e-8
DETR-Resnet	2.298e-8

Table 2. Results for Cats Image

AI Model	Objects	Xmin	Xmax	Ymin	Ymax	Area
Deformable-DETR	**Cat 1**	16.5	318.25	52.84	470.78	126113.395
	Cat 2	342.19	640.02	24.3	372.25	103629.9485
OwlVit-base-patch14	**Cat 1**	8.61	315.07	54.8	479.78	130239.3708
	Cat 2	335.98	637.93	14.93	373.28	108203.7825
OwlVit-base-patch32	**Cat 1**	1.46	315.55	55.26	472.17	130947.2619
	Cat 2	324.97	640.58	20.44	373.29	111362.9885
DETR-Resnet	**Cat 1**	13.24	314.02	52.05	**470.93**	125990.7264
	Cat 2	345.4	640.37	23.85	368.72	101726.3039
	Remote 1	40.16	175.55	70.81	117.98	6386.3463
	Remote 2	333.24	368.33	72.55	187.66	4039.2099
	Couch	-0.02	639.73	1.15	473.93	302461.005

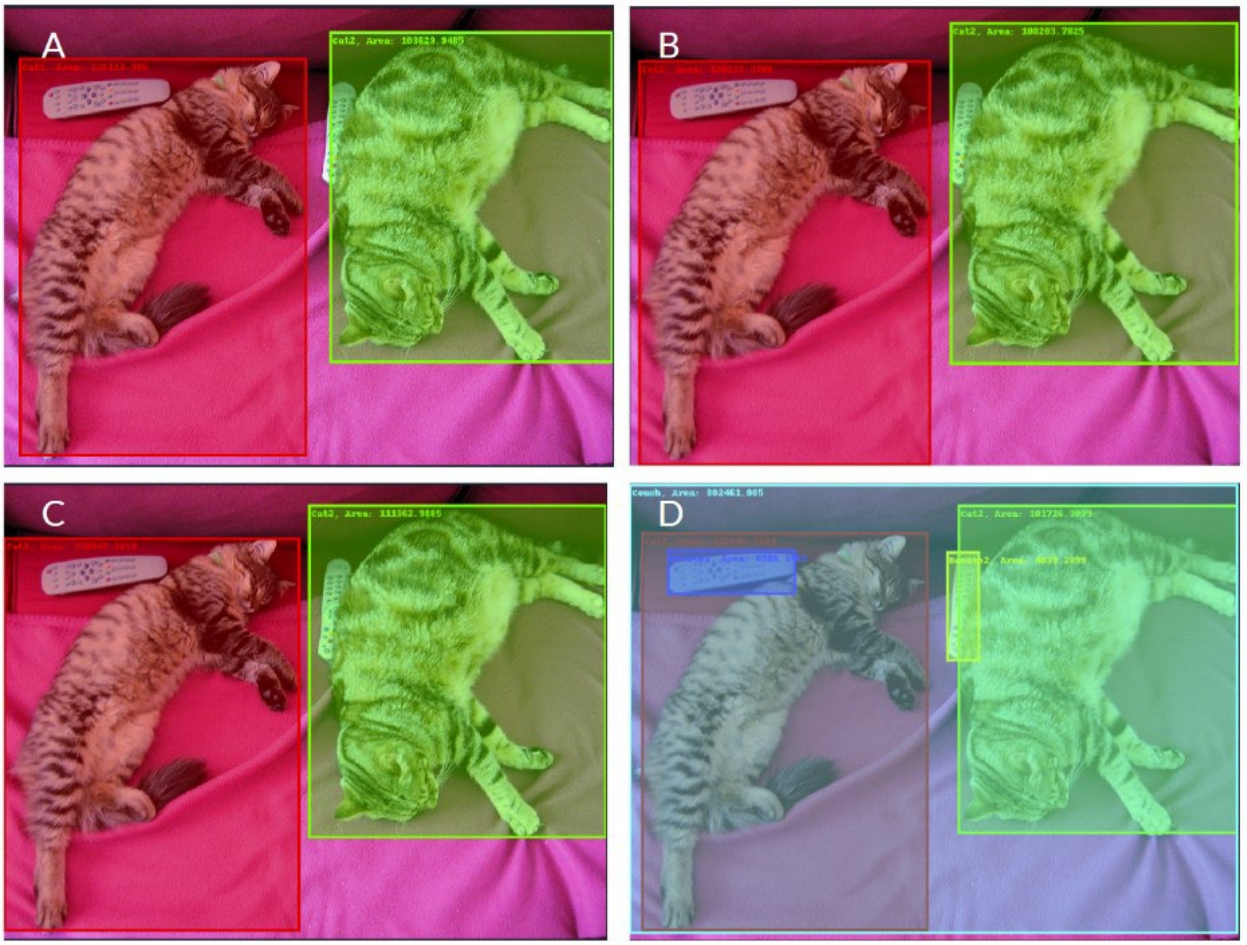


Figure 1: Visualization of results of each AI using the PyTorch code mentioned above; A = Deformable-DETR, B = OwlVit-base-patch 14, C = OwlVit-base-patch 32, D = DETR-Resnet.

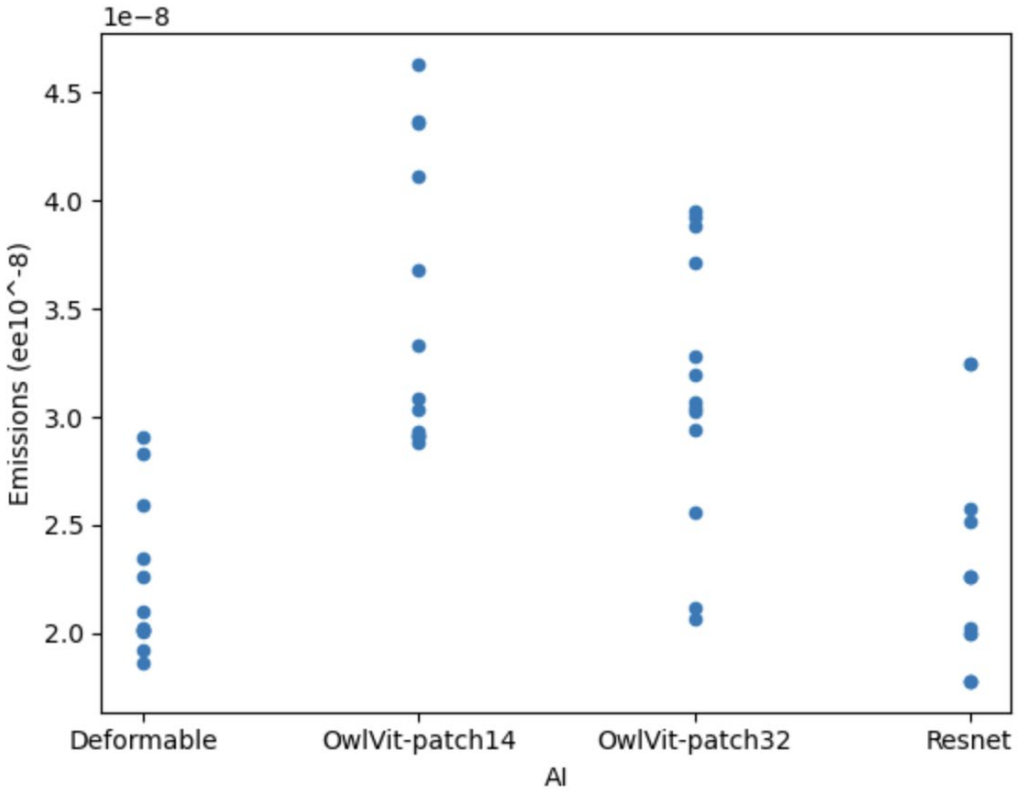


Figure 2: Carbon emissions of each AI (in kg). Each dot represents one test case.

Marching Band

In the following tables, Person 1-5 refers to the 5 people standing in the foreground of the image, from left to right. Person 6-10 refers to the 5 people standing in the background.

Table 3. Results for Marching Band Image

AI Model	Objects	Xmin	Xmax	Ymin	Ymax	Area
Deformable-DETR	**Person 1**	86.03	269.32	86.03	678.37	108569.9986
	Person 2	236.22	409.62	169.96	678.28	88142.688
	Person 3	374.97	555.99	144.92	675.71	96083.6058
	Person 4	529.94	727.86	108.9	678.44	112723.3568
DETR-Resnet	**Person 1**	81.9	270.39	157.91	679.92	98393.6649
	Person 2	239.44	412.45	164.01	680.3	89323.3329
	Person 3	374.82	558.33	140.49	680.33	999066.0384
	Person 4	513.99	725.44	108.24	680.3	120962.087
	Person 5	766.42	1022.96	89.05	679.07	151363.7308
	Person 6	112.28	174.96	145.22	264.53	7478.3508
	Person 7	497.38	550.67	146.26	216.13	3723.3723
	Person 8	689.72	753	171.95	413.34	15275.1592
	Person 9	32.54	131.16	119.49	421.01	29735.9024
	Person 10	-0.06	44.65	134.1	421.7	12858.596

Table 4. Average Carbon Emissions for Marching Band Image

AI Model	Emissions(kg)
Deformable	2.245e-8
OwlVit-base-patch14	2.636e-8
OwlVit-base-patch32	2.455e-8
DETR-Resnet	2.293e-8

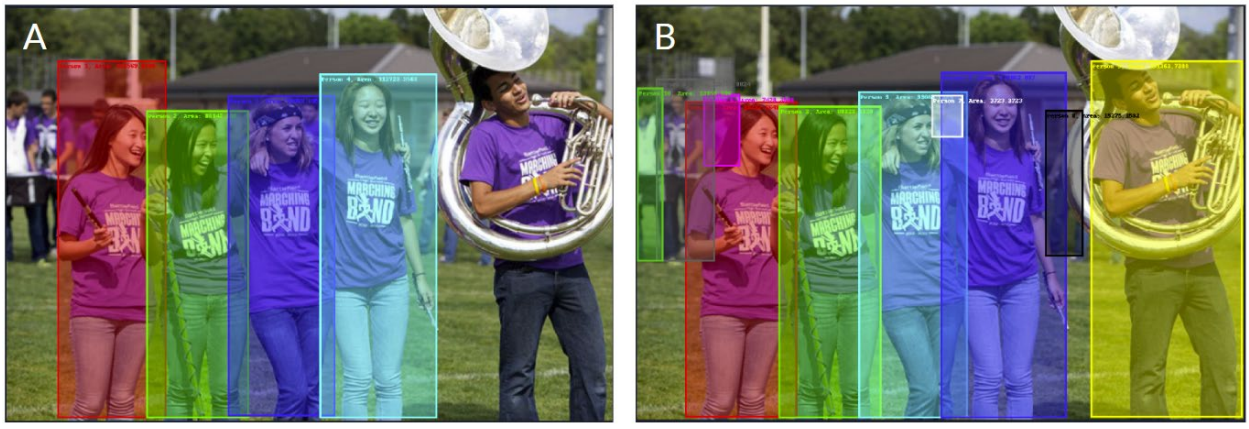


Figure 3: *Visualization of results of each AI using the PyTorch code mentioned above; A = Deformable-DETR, B = DETR-Resnet.*

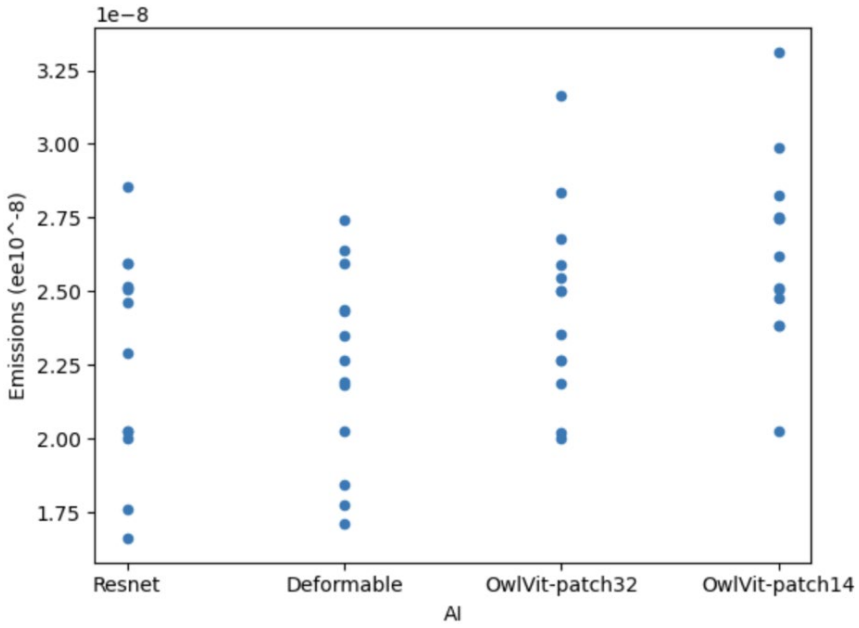


Figure 4: *Carbon emissions of each AI (in kg). Each dot represents one test case.*

Table 5. Remaining Test Images (A = Deformable DETR, B = OwlVit-base-patch14, C = OwlVit-base-patch32, D = DETR-Resnet, No Image = No Detection)

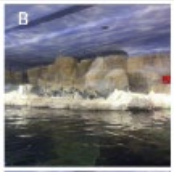
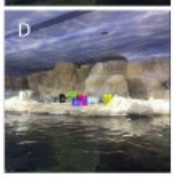
Image	Results
	
	 
	 

Table 6. Overall Average Carbon Emissions

AI Model	Emissions(kg)
Deformable	2.307e-8
OwlVit-base-patch14	2.654e-8
OwlVit-base-patch32	2.688e-8
DETR-Resnet	2.298e-8

Tables I and IV show the average carbon emissions of each AI model for the specified image. The average was calculated using the data in Figures 2 and 4 respectively.

Table V shows the results of the remaining images. Table VI shows the overall average carbon emissions of each AI across all five images.

Based on Table VI, the OwlVit-base-patch32 model generated the highest average carbon emissions, approximately 1.30% more than the OwlVit-base-patch14, 16.52% more than Deformable-DETR, and 16.97% more than DETR-Resnet.

The results from our test images and data indicate that the DETR-Resnet model performed the most consistently in terms of the number of objects detected among the four tested AI models. Additionally, it generated the least average carbon emissions. Moreover, the calculated area of each bounding box was smaller for the Resnet model while still effectively encompassing the entire object, indicating more precise results compared to the other models.

This insight highlights that larger models do not always yield the best results and that smaller models that generate lower carbon emissions can outperform their larger counterparts.

To further estimate the carbon emissions of larger and smaller object detection models when deployed in AI cameras in the United States, we conservatively assumed that the AI cameras would be continuously running object detection at 15 frames per second (fps) throughout the day (How many frames per second can the human eye see?, 2021). Assuming there are approximately 24 million households with video surveillance (Security Camera Statistics: 2022 Market Share Analysis & Industry Trends, 2023), and each household employing one AI camera, we can estimate the carbon emissions by multiplying the fps, the total number of seconds in a day (86,400), the number of AI cameras, and the average carbon emissions per run of the object detection software:

*fps * seconds * carbon emissions * number of cameras (2)*

It's important to note that the only variable that changes between each calculation is the average carbon emissions per run.

For this estimation, we considered the DETR-Resnet model as the smaller model and the OwlVit-base-patch14 as the larger model. Based on our experimental data, the average carbon emissions per run for the DETR-Resnet model and the OwlVit-base-patch14 model are 2.298183e-08 kg and 2.653695e-08 kg, respectively. Running the calculation as mentioned above, we find that the estimated total emissions for the DETR-Resnet and OwlVit-base-patch14 models are 714,826.840 kg per year and 825,405.293 kg per year, respectively. Thus, the larger model generated approximately 110,578.453 kg, or about 15.47% more carbon emissions. It is important to note that this difference will continue to increase as more cameras are installed and the fps of AI cameras is increased, making the carbon emissions savings of smaller models exponentially greater.

Language Translation

Our overarching objective for this section is to attain superior translation quality while minimizing the carbon emissions associated with these models. We have conducted an analysis wherein we noted the BLEU scores and model sizes for each utilized model. This investigation allows us to observe the correlation between employing smaller

models for a given task and the resultant reduction in carbon emissions. By leveraging these insights, we aim to strike a balance between translation quality and environmental impact.

The evaluation of translation models predominantly relies on BLEU(Evaluate custom models, 2023), an algorithm designed to assess the quality of natural language models. BLEU is among the pioneering metrics that claim a strong correlation with human judgments of quality and remains widely employed to evaluate text translation. A higher BLEU score indicates a higher level of quality for the model. Consequently, it has emerged as one of the most popular metrics for evaluating text translation.

In Table VII, we observe that the highest BLEU scores were produced by medium-sized models, not necessarily by larger models.

Table 7. Translation Models

Name	Type	BLEU Score	Size(GB)
facebook/wmt19-de-en	De-En	40.8	1.08
google/bert2bert_L-24_wmt_de_en	De-En	39.3	3.09
allenai/wmt19-de-en-6-6-base	De-En	38.37	0.266
facebook/wmt19-en-de	En-De	42.7	1.08
allenai/wmt16-en-de-12-1	En-De	25.75	0.235

For every translation task, we conduct ten separate runs using each model. These translation tasks involve passages ranging from 40 to 60 words in either German to English or English to German translation. To ensure environmental sustainability, we employ CodeCarbon (CodeCarbon, 2021) to track and record the carbon emissions generated by each test run. The CPU used to conduct these runs was an AMD Ryzen 7 5700U with Radeon Graphics. By collecting this data, we create plots that depict the total carbon emissions produced by each run, correlating them with the sizes of the respective models. Additionally, we plot the BLEU scores against the model sizes to observe the relationship between translation quality and model size.

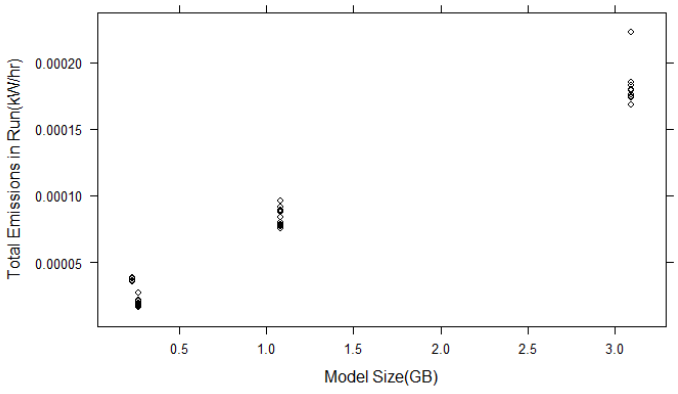


Figure 5:
Total Carbon Emissions in Relation to Model Size

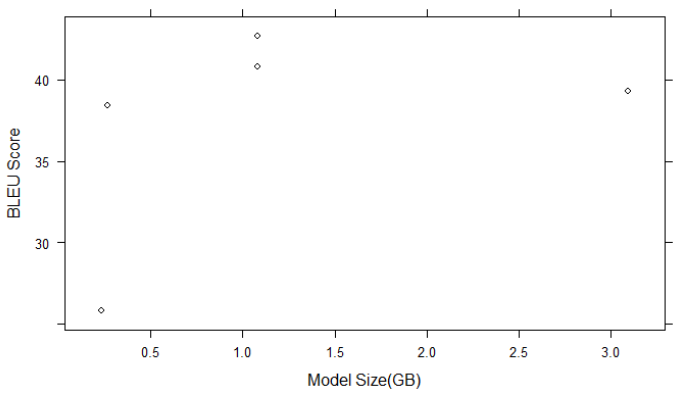


Figure 6:
Quality(BLEU Score) in Relation to Model Size

Figures 5 and 6 suggest a moderately strong correlation between carbon emissions and model size, indicating that larger models tend to generate higher carbon emissions. On the other hand, there is a weaker relationship between BLEU scores and model size, implying that increasing the size of the model does not necessarily result in a higher-

quality translation. However, it is important to note that larger models are more likely to produce higher rates of carbon emissions. Therefore, when considering translation tasks, it is crucial to strike a balance between model size, translation quality, and environmental impact.

Stable Diffusion

For the experiment conducted over Stable Diffusion, we used the Stable Diffusion WebUI created by AUTOMATIC1111 (AUTOMATIC1111, 2023). The laptop used had a 12th Gen Intel(R) Core(TM) i7-12700H CPU and an NVIDIA GeForce RTX 3060 GPU. This experiment was conducted in the central US region, in Austin Texas, where the self-deployment emission rate showed to be .00044 lb of CO2 per KHh.

One of the critical steps involved in the process was manually determining the sampling steps for generating the desired images. In Stable Diffusion, each sampling step aims to reduce noise within an initially noisy image and produce an output that aligns with the provided prompts. It is important to note that increasing the number of sampling steps directly leads to a proportional increase in the run time of image generation. Below are some indicative examples of the corresponding run times for reference:

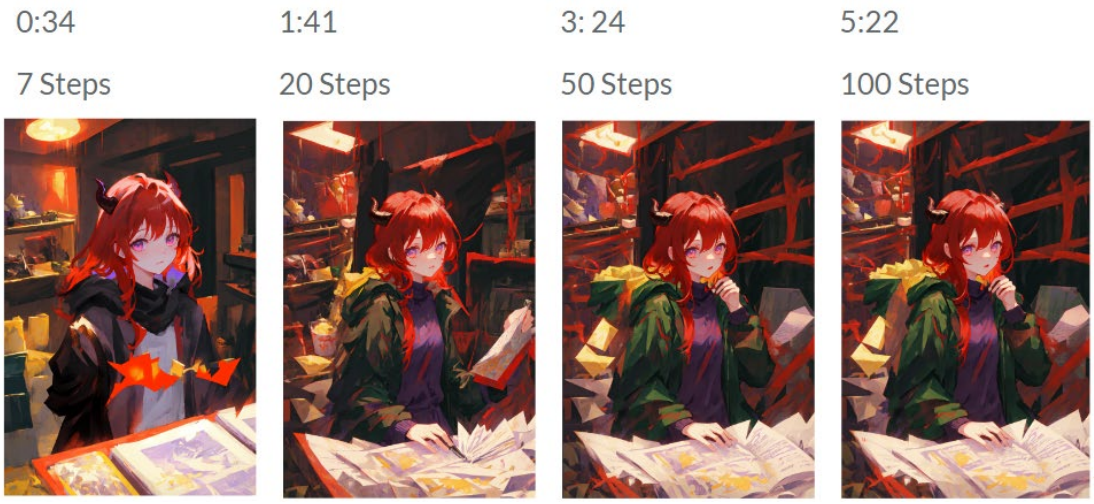


Figure 7: Stable Diffusion Performance by Step Count

Based on Figure 7., we can conclude that the quality of the generated images tends to improve as the number of sampling steps increases, up to around 20 steps. Beyond this point, there appears to be minimal improvement in image quality despite the continued increase in run time. Therefore, it seems that running the process with approximately 20 steps would be the most optimal approach. This balance allows for a reasonably high-quality output while also minimizing the additional run time required for further incremental improvements.

Now that we understood what the most optimal run steps are, we conducted four sets of 10 samples for each Checkpoint/vae. Each set was generated using one of the four sampling methods mentioned earlier. This approach allowed us to compare and evaluate the results of each method for different Checkpoints/vae.

Among the different sampling methods utilized, each provided a distinct set of outcomes. The DPM++ method resulted in the most detailed images, showcasing a higher level of visual intricacies and fine details. On the other hand, the Euler-a method yielded different outcomes, presenting unique characteristics and visual variations compared to the other methods. The specific details and differences between these sampling methods can be observed in the samples provided below in Figure 8:

Figure 8: Test Result of Various Sampling Methods

Below are test data results from different sampling methods. Figure 9 illustrates a trend where the best-performing method, DPM++, also has the longest average runtime. Following DPM++, Euler and LMS methods exhibit relatively shorter runtimes, and DDIM has the shortest average runtime among the methods considered. It's important to note that while DPM++ provides superior performance, it comes at the cost of increased computational time, while the others offer faster generation with potentially slightly lower-quality outputs. These factors should be considered when selecting the most suitable sampling method for image generation tasks.

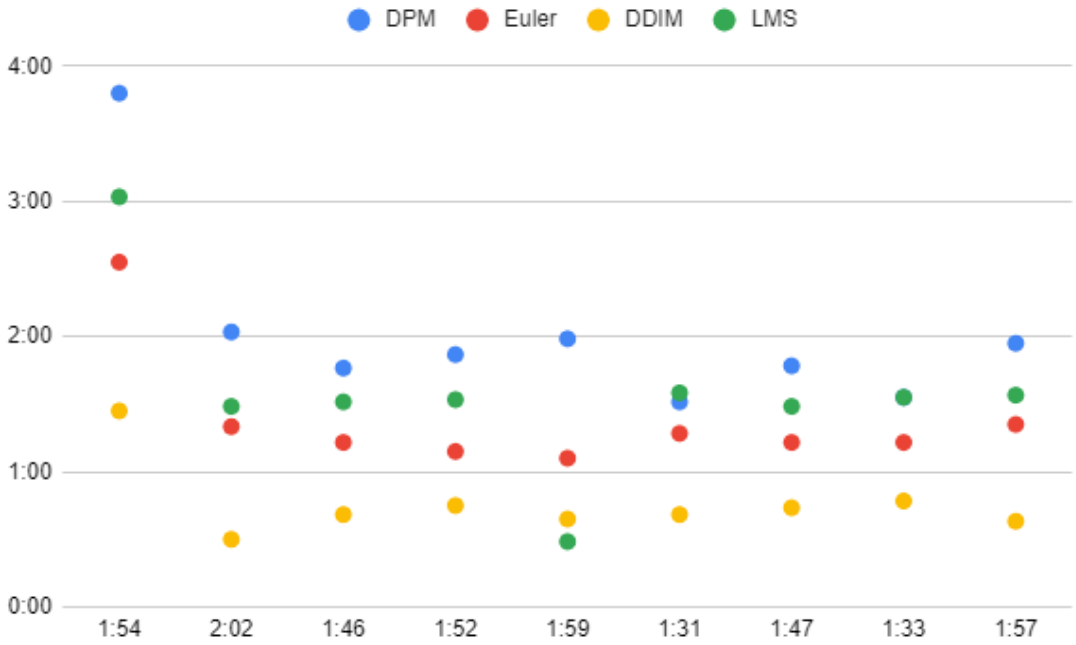


Figure 9: Test Result of Various Sampling Methods

The issue with image generation models lies in the fact that it is generally preferable for users to select the model that will produce the most desirable results. This means that even if using a smaller model helps in reducing carbon emissions, it may not be favored if it compromises the quality of the generated images.

In conclusion, when it comes to image generation models, selecting smaller models does not have a significant impact on the output quality. However, users can still optimize the settings and parameters for the chosen models to

achieve the best possible results. It is important to strike a balance between the desired image quality and the environmental impact when making such decisions.

V. Conclusions and Future Work

As the field of AI continues to advance rapidly, concerns about its environmental impact, specifically carbon emissions and global warming, are growing. The increasing demand for computing power in AI models is expected to contribute significantly to carbon emissions in the future. To mitigate this impact, we explored using smaller AI models that emit fewer greenhouse gases.

In our study, we conducted a comprehensive analysis of 12 AI models across different domains, including object detection, translation, and text-to-image generation. The results clearly demonstrated that smaller AI models have the potential to achieve comparable or even better performance compared to larger models, while drastically reducing carbon emissions.

However, it's important to acknowledge the limitations of our experiment. Firstly, the scope of tasks and AI models tested was limited, and our findings may not be generalizable to all AI models and applications. Additionally, hardware constraints prevented us from running larger models consistently, potentially impacting the accuracy of our results. Moreover, our measurement of carbon emissions relied solely on CodeCarbon, which, while widely used, may not be perfectly reliable or accurate. We also faced challenges in determining a standardized measure of quality, relying instead on subjective assessments, which introduces the possibility of bias in our conclusions. Finally, we observed that some AI models experienced a significant decrease in performance when using smaller versions, indicating the need for alternative solutions.

Moving forward, we plan to explore the potential of carbon-aware workload shifting as a means to reduce carbon emissions in AI models. By adopting this approach, we aim to achieve positive outcomes for AI models of all sizes and contribute to a global reduction in carbon emissions.

References

Abyssorangemix2 - sfw/soft nsfw. (2023, March 28). Civitai. https://civitai.com/models/4437/abyssorangemix2-sfwsoft-nsfw

AUTOMATIC1111. (2023, July 19). *Stable-diffusion-webui.* GitHub. https://github.com/AUTOMATIC1111/stable-diffusion-webui

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kegl, B. (2011). *Algorithms for Hyper-parameter Optimization.* In Proc. of NeurIPS. https://papers.nips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. (2020). *End-to-End Object Detection with Transformers.* ECCV 2020. Lecture Notes in Computer Science, vol 12346. Springer, Cham. https://doi.org/10.1007/978-3-030-58452-8_13

CodeCarbon. (2021). CodeCarbon. https://codecarbon.io/

Dettmers, T., & Zettlemoyer, L. (2019). *Sparse Networks from Scratch: Faster Training without Losing Performance?* arXiv preprint arXiv:1907.04840. https://doi.org/10.48550/arXiv.1907.04840

Dhar, P. (2020). *The Carbon Impact of Artificial Intelligence*. Nature Journal, Nat Mach Intell, 2, 423-425. https://www.nature.com/articles/s42256-020-0219-9

Evaluate custom models. (2023, July 19). Google Cloud. https://cloud.google.com/translate/automl/docs/evaluate

How many frames per second can the human eye see? (2021, March 2). CaseGuard. https://caseguard.com/articles/how-many-frames-per-second-can-the-human-eye-see/

Hugging Face models. (2016). Hugging Face. https://huggingface.co/models

Kasai, J., Pappas, N., Peng, H., Cross, J., & Smith, N. A. (2019). *Deep Encoder, Shallow Decoder: Reevaluating Non-Autoregressive Machine Translation.* arXiv preprint arXiv:2006.10369. https://doi.org/10.48550/arXiv.2006.10369

Lewis, P., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., & Stoyanov, V. (2020). *Leveraging Pre-trained Checkpoints for Sequence Generation Tasks.* Transactions of the Association for Computational Linguistics, 8, 264-280. https://doi.org/10.1162/tacl_a_00313

Mehlin, V., Schacht, S., & Lanquillon, C. (2023). Towards energy-efficient Deep Learning: *An overview of energy-efficient approaches along the Deep Learning Lifecycle.* arXiv preprint arXiv:2303.01980. https://doi.org/10.48550/arXiv.2303.01980

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., ... Houlsby, N. (2022). *Simple Open-Vocabulary Object Detection with Vision Transformers.* ECCV 2022, 728-755. https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136700714.pdf

Mukulsomukesh M. J. (2022, April 22). *How to draw bounding boxes on an image in PyTorch?* GeeksforGeeks. https://www.geeksforgeeks.org/how-to-draw-bounding-boxes-on-an-image-in-pytorch/

Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., Edunov, S. (2019). *Facebook FAIR's WMT19 News Translation Task Submission.* arXiv preprint arXiv:1907.06616. https://doi.org/10.48550/arXiv.1907.06616

Nishant, R., Kennedy, M., & Corbett, J. (2020). *Artificial Intelligence for Sustainability: Challenges, Opportunities, and a Research Agenda.* International Journal of Information Management, 53. https://www.sciencedirect.com/science/article/abs/pii/S0268401220300967

Pastel Mix Stylized Anime Model. (2023, May 24). CivitAi. https://civitai.com/models/5414/pastel-mix-stylized-anime-model

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., Luccioni, A., Maharaj, T., Sherwin, E. D., Mukkavilli, S. K., Kording, K. P., Gomes, C., Ng, A. Y., Hassabis, D., Platt, J. C., Creutzig, F., Chayes, J., & Bengio, Y.(2022). *Tackling Climate Change with Machine Learning.* ACM Computing Surveys, 55(2), Article 42, 1-96. https://dl.acm.org/doi/10.1145/3485128

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). *Green AI.* Communications of the ACM, 63(12), 54-63. https://dl.acm.org/doi/10.1145/3381831

Security Camera Statistics: 2022 Market Share Analysis & Industry Trends.(2023, July 15). OpticsMag. https://opticsmag.com/security-camera-statistics/

Stable Diffusion 2.1. (2023, July 4). Hugging Face. https://huggingface.co/stabilityai/stable-diffusion-2-1

Stable diffusion samplers. (2023, January 8). NightCafe. https://nightcafe.studio/blogs/info/stable-diffusion-samplers

Yigitcanlar, T., Mehmood, R., & Corchado, J. M. (2021). *Green Artificial Intelligence: Towards an Efficient, Sustainable and Equitable Technology for Smart Cities and Futures.* MDPI Sustainability, 13(16). https://doi.org/10.3390/su13168952

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., ... Dai, J. (2020). *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. ICLR 2021 Oral, September 2020. https://iclr.cc/virtual/2021/oral/3448