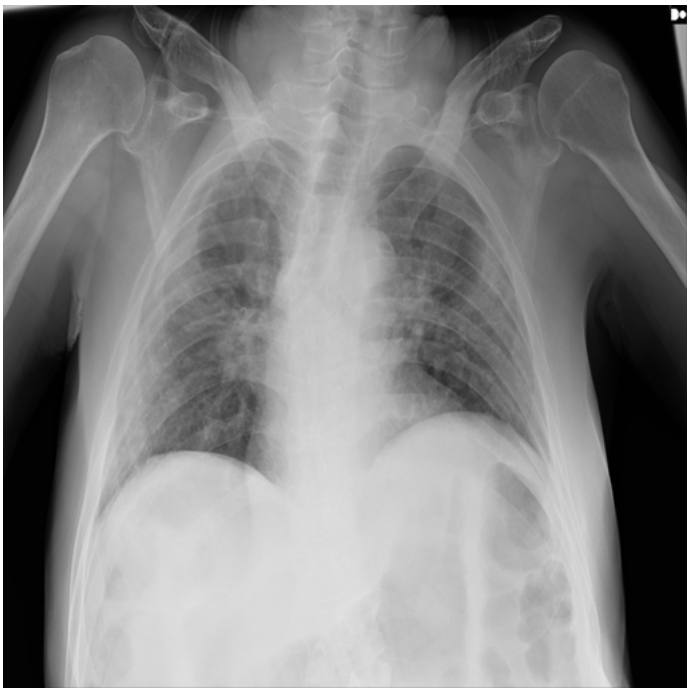
Carbon Footprint of Machine Learning Competitions With Medical Images

CHRISANNA KATE CORNISH CCOR@ITU.DK
DANIELLE MARIE DEQUIN DDEQ@ITU.DK

SUPERVISOR:
VERONIKA CHEPLYGINA
VECH@ITU.DK

6TH SEMESTER, SPRING 2023
DATA SCIENCE BSC
BIBAPRO1PE



IT UNIVERSITY OF COPENHAGEN

Carbon Footprint of Machine Learning Competitions With Medical Images

CHRISANNA KATE CORNISH
CCOR@ITU.DK

DANIELLE MARIE DEQUIN
DDEQ@ITU.DK

SUPERVISOR:
VERONIKA CHEPLYGINA
VECH@ITU.DK

***Abstract*—The increasing energy consumption and carbon footprint of machine learning (ML) with medical imaging due to high compute requirements has become a cause of concern. ML competitions have been shown to have diminishing returns, where gains are ever smaller alongside the training of thousands of models. Research is being done to evaluate the carbon footprint of individual models on both computationally expensive image and text datasets. In this work, we focus on the carbon footprint of not just a single model, but an entire ML competition where thousands of models are being trained. We present the results in order to highlight the carbon footprint of ML competitions. We use the tool CarbonTracker to find the estimated CO_2 emissions. We estimate the cost of the competition by evaluating the training cost of eleven different models on medical image datasets and extrapolating the results. With our work we hope to inform on the high energy cost of using the ML competition framework to solve medical image problems, and how this compares to performance improvements of the competition's winner. We further provide problems found in reproducing the results of the competition itself, reducing the practical applicability of such solutions.**

I. INTRODUCTION

MACHINE learning (ML) models, especially larger deep learning models that are trained on image or text datasets, can be computationally expensive to train. In addition, during development models are usually trained multiple times, for example to optimize hyperparameters. This combination can result in an increased energy consumption and therefore a large carbon footprint[1]. For example, Strubell et al. show that the carbon footprint of a commonly used model in NLP can be over 284,000,000g CO_2 [2]. For comparison, this is more than the CO_2 equivalent to sending three humans on a joy ride in space [3].

Kaggle is an online community and platform for ML and data science competitions [4] where many thousands of models are being trained. As of 17 April, 2023, Kaggle was hosting 108 image ML competitions. A recent study looked into eight Kaggle medical imaging challenges [5]. In these competitions there were over 58,000 total entries listed on their private leaderboard. This would indicate the training of more than 58,000 models on medical images for just eight of the 108 image competitions hosted on the platform.

The winners of Kaggle challenges do not seem to make significant results. In that same study by Varoquaux et al. they found that in only two of those eight competitions did the winning method make significant improvements compared to the evaluation noise [5].

With the competition framework encouraging such a high volume of models being trained, while exhibiting ever smaller performance gains, we wanted to quantify the carbon footprint of an entire Kaggle competition. There are some recent efforts to quantify the carbon footprint for training models on medical image data [6] as well as text data [2]. However, we were not able to find studies evaluating Kaggle competitions.

In this study we investigate the carbon footprint of a Kaggle competition by calculating a lower and upper bound cost. The lower bound is based on the cost of training a simple model and the upper bound is based on a complex ensemble requiring multiple models. To make this quantification we evaluate publicly available code from one of the submitting teams to get the carbon footprint of a large model being trained. We get further data by evaluating an external ML project that uses transfer learning to train smaller models on medical image datasets. We use these results to compute the "per-submission" carbon cost estimation, and extrapolate these results to get the total cost of the competition.

We found the range of the estimated cost of the SIIM-FISABIO-RSNA COVID-19 Detection competition to be between 50,000 - 180,000,000g CO_2. For comparison, the upper bound is equivalent to 40 gasoline-powered passenger vehicles driven for one year [7].

II. RELATED WORK

Computer vision has been shown to have advanced hardware requirements. Computer vision is a form of ML

[1]Carbon footprint is a common term to express the direct and indirect greenhouse gas emissions, particularly of carbon dioxide, during some activity [1]. It is usually presented as a weight of CO_2 or CO_2 equivalent.

that trains models on image, video, and other visual forms of input, and requires a lot of data. One such technology used to accomplish computer vision is a form of ML called deep learning (DL) [8]. DL methods are highly effective due to advances in high-tech central processing units (CPUs) and graphics processing units (GPUs) as well as the availability of a huge amount of data during training [9].

Compute requirements of DL have a high carbon footprint. The use of CPUs and GPUs in DL has a high energy requirement, which increases the demand for energy production [6]. In 2010 energy production was shown to be responsible for approximately 35% of total human-caused green house gas emissions [10].

Medical imaging applications of DL are also computationally expensive. A high success rate of such applications similarly depends on a large number of datasets [11]. Medical image datasets can be comprised of very large data, for example, 3-D images. Creating realistic 3-D computational models of complex biological structures can be a computationally expensive process, requiring the use of GPUs and multi-core CPUs [12].

A. Research Analysing ML Carbon Footprint

Measuring the carbon footprint of training a model requires understanding of the emissions of energy grids. Work has been done to compare regional energy grid differences, and has found there to be large variation in CO_2 emissions of training a model depending on the location [13].

There are some recent efforts to quantify the carbon footprint for training ML models. For example Strubell et al. quantify the approximate financial and environmental costs of training a variety of current models used in NLP [2]. In their study they "evaluate the full computational resources required for development and tuning of a recent state-of-the-art NLP pipeline". They evaluate different base models such as a Transformer model and ELMo, and track details such as GPU used, run-time, and training steps. In their research they report findings of, for example CO_2 emissions and energy usage (kWh). They show that training a model in NLP can range from around 11,000g to over 284,000,000g CO_2.

Further work has shown that models trained on image datasets are also computationally expensive. Selvan et al. provide a benchmark carbon footprint of ML methods for medical image analysis (MIA), and compare the features of four tools used to quantify the carbon footprint of ML models trained for MIA [6]. In their research they train a base model on three diverse MIA datasets for segmentation. They note hardware used such as GPU and CPU, and track CO_2 emissions (kg), equivalent distance travelled by car (km), and energy usage (kWh). In addition, they compare the results across three geographic locations. Their results show that CO_2 emissions range from 26,000 to 58,000g per model.

Various studies have given recommendations to minimize or track carbon footprint for ML researchers. Strubell et al. suggest reporting training time, computational resources required, and model sensitivity to hyperparameters [2]. They further suggest that research done in both industry and academia prioritize computationally efficient hardware and algorithms.

Several works have been done to address efficient model training. Dettmers et al. developed an algorithm that accelerates training of deep neural networks that maintain sparse weights throughout training while achieving dense performance levels [14]. Lan et al. developed a version of BERT with lower memory consumption and an increased training speed [15]. A recent literature review compares DL techniques for MIA while reporting training and execution time [16].

B. Tools For Analysing Carbon Footprint

There are a few tools available for measuring the energy consumption or carbon footprint of training ML models. The *Machine Learning Emissions Calculator* is one such tool that offers an easy-to-use online calculator [17]. With this tool a user can input the hardware, runtime, cloud provider and global region used when training a model, and it will output approximate emissions. However, this tool does not factor in CPU use, and is unable to track energy usage in real time during training.

Another tool, and the one we use in our research, is CarbonTracker [18]. This tool is able to be used for tracking and predicting the energy consumption and carbon footprint of training deep learning models. The user is able to track a single epoch of training, and the prediction capability will estimate the energy consumption for the number of epochs specified. Therefore this tool can be used track the full cost of training a model, without needing to re-run the full length of training. Both CPU and GPU, as well as location are taken into consideration for the calculation.

C. Research Evaluating Kaggle

In the research by Varoquaux et al., they evaluate *algorithmic improvements* of Kaggle competitions, and show that a strong focus on benchmark performance can lead to *diminishing returns* [5]. *Diminishing returns* is where increasingly large efforts achieve ever smaller performance gains. To measure this, they quantified the *evaluation noise* - the spread of performance differences between the public and private test sets. They find that the performance of submissions is often a byproduct of where the data were split for the public and private test sets, and not a true evaluation of the submitted models. This distribution was compared against the *winner gap* - the difference in performance between the winner and the top 10% team.

D. The Gap That Our Research Fills

While much research addresses different aspects needed to calculate the CO_2 emissions of training ML models, there is not work that specifically addresses the carbon footprint of Kaggle competitions. With a high volume of models trained per competition, and a track record of having diminishing returns, our work looks into the carbon footprint of Kaggle competitions in order to address this gap and inform the scientific community of a pitfall in the competition format.

III. Model/Datasets

We measured the carbon cost of training two models, NFNet and ResNet50, on seven different medical image datasets.

- NFNet Datasets:
  - SIIM 2021
  - NIH Chest X-ray
- ResNet50 Datasets:
  - ISIC
  - Breast
  - Chest
  - Knee
  - Thyroid

A. NFNet Model and Datasets

NFNet, short for Normalizer-Free ResNets, is a family of classifiers based on the ImageNet dataset that achieves comparable accuracy to other state-of-the-art (SOA) models while reducing training time [19]. Previous SOA works in DL use batch normalization, which is shown to have certain benefits, including a regularizing effect due to noise in batch statistics [20]. NFNet does not use batch-normalization and instead uses a few techniques including adaptive gradient clipping. Typically gradient clipping is used to prevent exploding gradients during back-propagation; this is when the gradients get too large during training. Adaptive Gradient Clipping will modify, or clip the gradient if the gradient-to-weights ratio exceeds a given threshold [21].

We chose to evaluate the NFNet model and corresponding datasets because they were in the solution submitted by the 2^{nd} place team [22] in the private leaderboard of the SIIM-FISABIO-RSNA COVID-19 Detection Kaggle competition [23] that we evaluated. Their training procedure includes three parts: two pipelines and a final detection stage. They train a total of twenty-five models.

In *pipeline one* they train an ensemble of NFNet models using the resized SIIM 2021 [24] dataset as well as the NIH chest X-ray dataset [25]. The resized SIIM 2021 dataset is a version of the original SIIM 2021 dataset [26] used in the competition, and consists of 6,334 chest X-rays. The NIH dataset consists of over 112,000 chest X-ray images from more than 30,000 unique patients.

We evaluated only *pipeline one*, which includes pre-training and training an ensemble of fourteen NFNet models. This gave us the training cost of one large ensemble to estimate the upper bound cost of the competition. *Pipeline two* and the *detection* phase had broken branches on GitHub, and incomplete explanation on how to run the code. Therefore these stages were unable to be ran and evaluated.

B. ResNet50 Model and Datasets

The ResNet50 architecture is commonly used in medical imaging and has been shown to perform well in image classification [27]. The ResNet50 model we evaluated and respective datasets came from source code in research by Juodelyte et al. [28]. In this research the ResNet50 models were initialized using ImageNet weights, and models were trained with both the base weights frozen and not frozen.

We evaluated the training of ResNet50 models based on ImageNet weights using five of the eight datasets available in the research by Juodelyte et al. These include the ISIC dataset of lesion images [29, 30], as well as Breast ultrasound [31], Chest X-ray [32], Knee MRI scan [33], and Thyroid ultrasound [34] datasets.

This research provided ten additional models. We evaluated the cost of training on each of the five datasets twice; once when only the classification layer is trained (base weights frozen), and again when training both the classification layer and the model weights (base weights not frozen).

We needed to supplement our data by evaluating the cost of training additional models using medical image datasets that were external to the competition. Within the competition only nineteen submitting teams have publicly available code, and of those many were unable to be ran or would take substantial work to get running. In addition, we needed the cost of training simpler models to calculate the lower bound estimation for the cost of the competition.

C. Kaggle Metadata

The Kaggle Metadata [35] are comprised of over thirty tables containing information on Kaggle competitions, including details about the competitions, participants, teams, submissions, forum posts, and other relevant data. The four tables we used in the analysis are shown in Table I.

Table	Rows	Columns
CompetitionTags	793	3
Competitions	5,576	42
Teams	6,180,319	14
Submissions	12,130,309	11

TABLE I: Kaggle metadata tables that we used during analysis. Includes number of rows and columns within each table. [35]

These four tables were combined to perform exploratory analysis. We isolated the “image” competitions first, and

considered different types of competitions. We also separated and analyzed the submissions for the SIIM-FISABIO-RSNA COVID-19 Detection competition.

IV. Method

A. Stage One: Initial Choices

Fig. 1 shows the steps we follow for stage one, our approach to the initial setup. We select the competition category to look though, select a specific competition, find a team who participated, then finally select a tool to evaluate the carbon footprint.

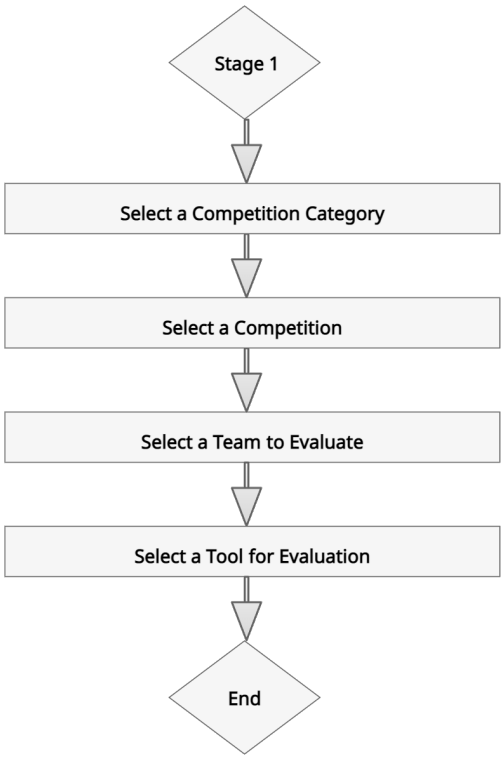


FIG. 1: Stage one, showing our approach to the initial setup. We select the competition category to look though, choose a competition, find a team who participated, then select a tool to evaluate the carbon footprint.

Select a Competition Category: We looked for competitions within the category *code* competition that use medical image data. On Kaggle, a *code* competition requires that competitors submit their solutions inside a Kaggle Notebook to participate. This would provide us the code needed for us to evaluate.

This category is intended to balance out the competitions with all users having the same hardware allowances. This should make it easier to evaluate the carbon footprint of just a few teams while making a claim about the carbon footprint of the entire competition.

Select a Competition: We then searched for a relatively recent competition and found the SIIM-FISABIO-RSNA COVID-19 Detection competition [23]. A recent competition provides a number of benefits. The code provided by the participants would not have deprecated dependencies. Both their code and the data would have a higher chance of being available publicly. Furthermore, this specific competition was one of the eight analyzed by Varoquaux et al. [5]. This was ideal, as we were interested in comparing the carbon footprint of our chosen competition against the presence of *diminishing returns*.

Select a Team to Evaluate: Within this competition we looked at the top teams in the private leaderboard who have publicly available code. The private leaderboard is based on a team placement after evaluation on a private test set of the data, and the placement here determines the winner of the competition. We went through the leaderboard until we found a team with code that we could reproduce, in this case the 2nd place team [22].

Select a Tool for Evaluation: The next step was to select a tool to use in our analysis to evaluate the carbon footprint, and we chose CarbonTracker [18]. Listing 1 shows an example output of CarbonTracker, with the actual consumption measured for one epoch of training as well as the predicted consumption for five epochs.

This tools was ideal for our research. The values tracked, and subsequently predicted are typical for the field [2, 36] and would give us a good comparison. These values are training time (HH:MM:SS), energy usage (kWh), CO_2 equivalent (g), and distance traveled by car (km). Also, CarbonTracker would facilitate us minimizing our own carbon footprint because it is able to run on a single training iteration and predict the cost of the full training without it having to be ran.

```
CarbonTracker:
Actual consumption for 1 epoch(s):
        Time:   0:52:31
        Energy: 0.170745 kWh
        CO2eq:  6.787123 g
        This is equivalent to:
        0.063136 km travelled by car
CarbonTracker:
Predicted consumption for 5 epoch(s):
        Time:   4:22:35
        Energy: 0.853726 kWh
        CO2eq:  33.062484 g
        This is equivalent to:
        0.307558 km travelled by car
CarbonTracker: Finished monitoring.
```

LISTING 1: Example output of CarbonTracker. It measures the actual consumption of training one epoch. It predicts the cost for the full training, which in this case is five epochs. The values tracked are time (HH:MM:SS), energy (kWh), CO_2 emissions equivalent (g), and distance traveleld by car (km).

B. Stage Two: Calculate Carbon Footprint

Fig. 2 shows the steps of our second stage, where we calculate the carbon footprint of the competition. We evaluate

the cost of training a single model, calculate the whole competition cost, and then calculate the carbon footprint of our research.

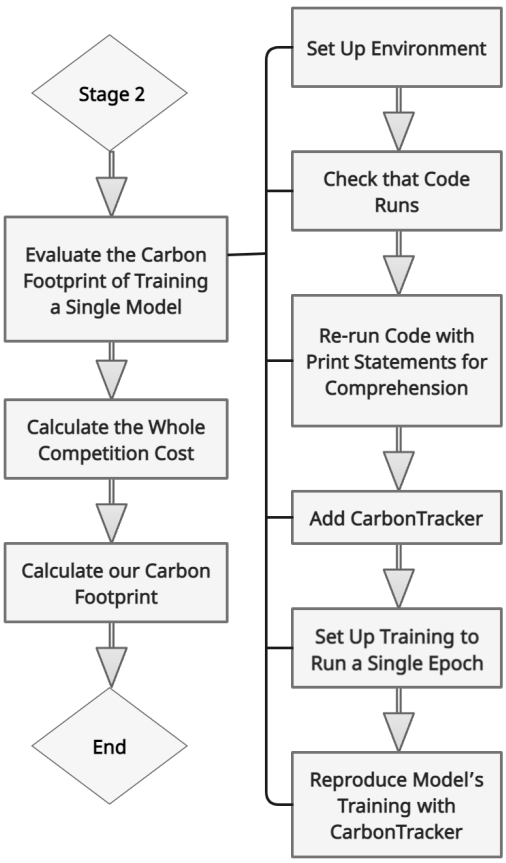


FIG. 2: Stage two, the steps we took to calculate the carbon footprint.

Evaluate the Carbon Footprint of Training a Single Model: The following are the steps we took for setting up and evaluating the training cost of a model using CarbonTracker:

1) Establish necessary environment for running the code.
2) Check that training runs without CarbonTracker added and that no issues occur. Debug as needed.
3) Comment out training and add print statements to further understand epochs, etc needed for training. This is essential for setting the maximum number of epochs for CarbonTracker to estimate the total cost.
4) Add CarbonTracker to the code and run a short test with a forty-five minute time limit. Any issues should appear within this time to verify if CarbonTracker was set up correctly.
5) Set up training to run with a single epoch.
6) Reproduce a single epoch of the model's training procedure with CarbonTracker added and use that to estimate the full training cost.

We ran the previous steps to evaluate the cost of training eleven models, one being the NFNet ensemble of *pipeline one* from the 2nd place team. The remaining ten were ResNet50 models trained on different datasets in the transfer learning research by Juodelyte et al. [28].

Care was taken to try to minimise the carbon footprint of our research. For example, we submitted jobs with time limits using SLURM, an open-source job scheduling system [37]. This would allow us run any debugging without having to run the full training. Also, by setting up training to run with a single iteration, we were able to predict the full cost of training without having to run the full training.

The estimated carbon footprint of training a model for a single submission is calculated based on a range. The lower bound is based on the lowest cost of training a single model once. The upper bound is based on the highest cost of training a single model out of the eleven models we evaluated.

Calculate the Cost of the Whole Competition: This figure is also given in a range, with a lower and upper bound. To calculate this we first look at the total number of submissions, which is 32,307 for all 1,305 teams. Inspired by Varoquaux and Cheplygina's paper [5] in order to take into consideration those who did not participate fully, we only count teams who submitted at least three times. This brings the number of submissions down to 31,751. We use this figure to calculate the lower and upper bounds by multiplying by the lowest and highest single submission costs respectively.

Calculate Our Carbon Footprint: We kept track of the number of times we trained each model, and how much of the training had completed. For example, we ran the training of the ResNet50 model on the Thyroid dataset five times but only once on the Knee dataset during development and testing. Therefore we multiply the cost of the first model by five, and the cost of the latter by one when calculating our total cost.

C. Explore Metadata

To help us select a competition category, we analyzed the distribution of submissions in each competition within Kaggle metadata [35].

First we isolated "image"-tagged competitions from the data. To do this we joined the *Competitions* and *CompetitionTags* tables on the "Id" and "CompetitionId" fields. This eliminated around 86% of the competitions. Some reasons for this might be that competitions are not tagged, or use non-standard tags that are not included in the *CompetitionsTags* table.

When filtering the data to only contain competitions with the "image" tag, this gave us 108 competitions to work with. We chose to filter by this tag due to our interest in evaluating medical image competitions. This tag was also the most relevant one used for the SIIM-FISABIO-RSNA COVID-19

Detection competition that we evaluated, along with other similar competitions such as the 2017 Data Science Bowl [38].

Of the "image"-tagged competitions we then selected those which had a deadline date from 2019 onwards. This is when *code* competitions were introduced. This narrowed down the competition list to sixty-six.

We then wanted to do exploratory analysis on these remaining competitions. Out of the sixty-six, thirty-six competitions were *code* competitions and the remaining thirty were *traditional* competitions. We compared the submission rates between these two categories to see if there were differences. Fig. 3 shows that *code* competitions generally have higher average submissions per team than *traditional* competitions.

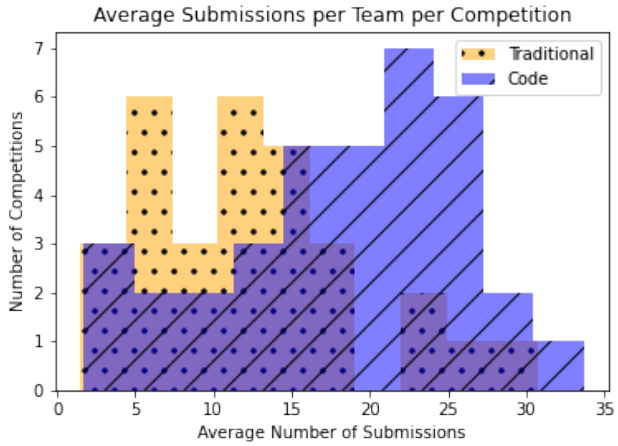


FIG. 3: Histogram showing the average number of submissions per team in both *code* and *traditional* competitions. The x-axis shows the average submissions per team for each competition, and the y-axis shows the number of competitions.

We further looked into these two competition categories to see if other differences exist. We looked at the minimum and maximum number of submissions, and found that *traditional* competitions had between 27 and 101,845 submissions, whilst *code* competitions were between 767 and 81,524. Looking at the box plot in Fig. 4, we can see there are generally a lot less submissions for the *traditional* style competitions.

The differences in the activity levels of these two categories confirmed our decision to further investigate one category individually.

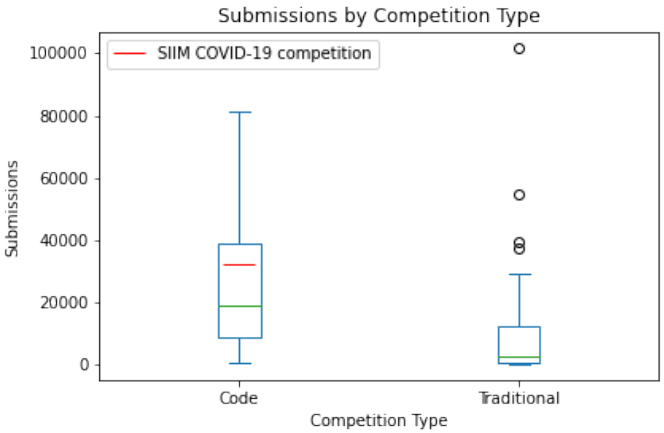


FIG. 4: Box plot comparing the total submissions on the y-axis in both *code* and *traditional*. The number of submissions for the SIIM-FISABIO-RSNA COVID-19 Detection competition is shown in red, clearly within the interquartile range.

V. RESULTS

A. There Is Variation in the Carbon Footprint of Training Individual Models

We show that the cost of training an individual model once ranges from 1.58 to 5752.78g CO_2 equivalent, as shown in Table II. In terms of distance driven by a car, this is equivalent to 0.02 to 53.51 kilometers.

Model	Dataset	Freeze	CO_2 (g)	Distance (km)
NFNet	NIH and SIIM	N/A	5752.78	53.51
ResNet50	ISIC	True	7.60	0.07
ResNet50	Breast	True	5.05	0.05
ResNet50	Chest	True	13.12	0.12
ResNet50	Knee	True	4.78	0.05
ResNet50	Thyroid	True	1.58	0.02
ResNet50	ISIC	False	120.22	1.12
ResNet50	Breast	False	12.57	0.12
ResNet50	Chest	False	36.43	0.34
ResNet50	Knee	False	18.38	0.17
ResNet50	Thyroid	False	1.75	0.02

TABLE II: Carbon footprint per model, shown in both CO_2 emissions in grams and distance driven in kilometers. The model column shows the base model used during training and dataset shows the respective dataset. The freeze column indicates whether the base model weights were frozen during training. The NFNet model did not have this parameter.

B. The Carbon Footprint of a Kaggle Competition Is Estimated To Be Greater Than the Emissions of Sending a Single Human Into Space.

We show that the estimated carbon footprint of the SIIM-FISABIO-RSNA COVID-19 competition as a whole ranges from 50,262 to 182,656,518g CO_2 emissions. This is shown in Table III. This emissions cost is equivalent to 476 to 1,698,996 kilometers driven by a car - approximately forty-two times around the planet[2].

[2]The Earth's circumference is approximately 40,000km

	Total CO_2 (g)	Total Distance (km)
Lower Bound	50,262	476
Upper Bound	182,656,518	1,698,996

TABLE III: This table shows the estimated carbon footprint of the SIIM-FISABIO-RSNA COVID-19 competition. The results include the upper and lower bound in both CO_2 emissions (g) and the equivalent distance driven by car (km).

According to the 2022 World Inequality Report an eleven minute space flight emits at least seventy-five tonnes of CO_2 per passenger [3]. Therefore, this particular Kaggle challenge has an upper bound estimated carbon footprint that is more than the minimum emissions of sending two people into space.

The results seem significant when compared to familiar consumption. Fig. 5 compares the lower and upper bound estimated CO_2 emissions of the competition to familiar use [2]. The data for this table can be found in Appendix A. As shown, the lower bound estimation is less than the CO_2 emissions of a single passenger travelling by plane between New York and San Francisco. This carbon footprint is notable, but not incredibly significant if it means helping to solve an important medical issue. However, the upper bound is more than three times the CO_2 emissions that a car emits over its entire lifetime, or thirty-six years of the average human life.

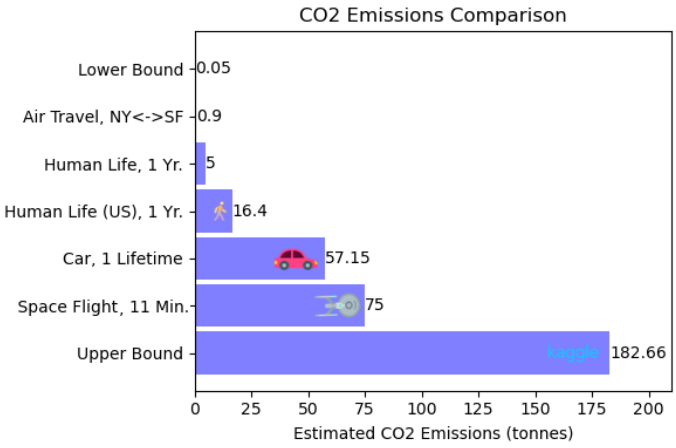


FIG. 5: This shows the CO_2 emissions in different categories, including familiar activities. The lower and upper bounds are our estimated results of the Kaggle competition. Air travel cost is for one passenger. Both human life categories are an average over one year. The car category is an average over a car's lifetime, including fuel. The eleven minute space flight is the minimum cost for sending one passenger into space. [3, 39, 40].

C. Our Carbon Footprint Is Low, but Not as Low as It Could Be

Our carbon footprint is almost double what it could be. In Table IV we show our carbon footprint to be 3100g CO_2 emissions or the equivalent to twenty-nine kilometers driven by car. This is based on the not-perfect world of model development, where we had to run some sections of training more than once. If everything ran smoothly without needing to fix any problems along the way, our CO_2 emissions would be 1483g, the equivalent to fourteen kilometers driven by a car.

CO_2 (g)	Distance (km)
3100	29

TABLE IV: This shows our total carbon footprint. It was calculated by multiplying the cost of each stage of training by the number of times we ran that stage during development of this project.

The result of our carbon footprint is insufficient. There is more that goes into evaluating the carbon footprint of ML than tracking the training of a model. In our study, we used a high performance cluster (HPC) for data storage and executing code, and used personal laptops for development and testing. The costs of activities outside of training a model were not tracked, and subsequently not added into our total carbon footprint.

D. It Is Difficult to Estimate the Carbon Footprint of a Kaggle Competition

We are not able to make a precise calculation due to missing data. For example, only nineteen teams have publicly accessible code. While we can see that these teams used large ensembles, we do not know what sort of models the other 1286 teams trained. Also, there is a lot of energy usage from other aspects outside of training a model, such as data storage, and the development and testing of a model [41]. We count each submission as a model being trained, but we do not take into consideration the carbon footprint of the other tasks involved.

Our Estimations Could Be Low: All of our code was run in Denmark, which could lower our estimations. The carbon footprint of training a model depends on the energy grid it is run on. Henderson et al. show that emissions can be reduced by up to thirty times just by running experiments in locations powered by more renewable energy sources [13]. All our scores are calculated based on Copenhagen, which is an energy efficient location. To compare, one study showed that the carbon emissions in Estonia are easily three times higher than in Denmark [5]. Kaggle competitions are open to a world-wide audience, and as such the potential variation in the carbon footprints could be high. Given this, the predicted carbon footprint may be underestimated.

Our estimation for the upper bound cost of the whole competition may be low. We were only able to track the energy cost of *pipeline one* during our calculation for the cost of the 2nd place team. This included fourteen out of the twenty-five models they used during training, meaning we tracked around half of their models. Since the upper bound cost of the competition is based on the cost of training this pipeline, this has lowered the calculation for the upper bound.

Our estimation for the cost of the lower bound may also be conservatively low. The other ten models we evaluate

are only training a single model instead of an ensemble, and use transfer learning with pre-trained weights on a large dataset, which is shown to have a reduction in training time as well as energy consumption [42]. The other competitors in the top ten of the competition also trained an ensemble of models in a similar way to the team we evaluated.

Our Estimations Could Be High: The upper bound estimation for the whole competition may be high. We use the number of submissions as a proxy for the number of times a model was trained. For example, the 2nd place team submitted 170 times. It could be that they ran the notebook 170 times because it required some debugging that was less computationally expensive than training a model the same amount of times.

Our estimation could also be high due to the time of day that we ran training. One study showed that "a model trained during low carbon intensity hours of the day in Denmark may emit as little as one-fourth the CO_2 emissions of one trained during peak hours" [5]. We were using a HPC to do our evaluations and had to submit jobs to a queue. Therefore we did not have control over the time of day that training occurred, and were not able to select a more energy efficient time.

E. There Are Problems Regarding Code Reproducibility

Medical imaging solutions submitted to Kaggle competitions may not be able to be practically applied due to problems with reproducibility. Previous studies have looked into problems with code reproducibility [5, 43, 44]. Some such issues are insufficient documentation, differences in operating systems and packages, missing implementation details, and lacking publicly available frameworks.

Problems we ran into are:

- Lack of comprehensible instructions.
- Lack of proper requirements files.
- Variation in local environment setups.
- Typing errors and other mistakes present in code.
- File paths that need to be modified are not easy to find for customization.
- Discrepancies in variable names throughout the source code.
- Datasets no longer available.
- Necessary branches in a git repository being broken.
- Unexplained parameter options.
- Missing implementation details.

These issues are hurdles in both carbon footprint assessment as well as applying the solutions to practical use. For evaluation, considerable time and effort was spent trying to get just *pipeline one* of the 2nd place team to run. We had also tried getting the 1st place team's submission to run, and were unable to do so. Our estimations would have been more accurate if we were able to evaluate more submissions. Furthermore, problems with code reproducibility could hinder the process of creating practical applications of these solutions; in this case being used as a tool to help medical professionals to detect and diagnose the presence of COVID-19.

F. Code Competitions Do Not Balance the Competition

We find that *code* competitions on Kaggle are not as fair as they are intended. The SIIM-FISABIO-RSNA COVID-19 Detection competition has a specific run-time limit of less than or equal to nine hours for either a CPU or GPU notebook.

The 2nd place team that we evaluated far exceeded that time limit by training externally and uploading weights. Their training procedure had three stages. In the first stage, *pipeline one*, there were two steps; pre-training and training. When evaluating their pre-training step alone, it took more than five days using sixteen CPU cores and an A100 40GB GPU. The training then took over seventeen hours using sixteen CPU cores and two A100 40GB GPUs. This was for a single epoch (out of five) at each step. This does not take into consideration the other two stages which we were unable to evaluate, or the additional time used in the submitted notebook itself.

The other teams in the top ten used a similar procedure; training an ensemble of models externally and uploading their weights. In doing so, participants are able to bypass the time and hardware limits that are imposed by the requirements that attempt to balance the competition.

VI. Discussion

A. Something Can Be Done To Prioritize a Focus on the Carbon Footprint of a Kaggle Competition

Without a focus on the carbon footprint it is challenging to evaluate, causing variation in the results. The upper bound estimated cost of the SIIM-FISABIO-RSNA COVID-19 challenge is more than the emissions equivalent of a wealthy human taking a leisurely ride in space. If this value was certain, it would be a cause for concern. The lower bound, however, is almost half of the carbon footprint of a single passenger flying from New York to San Francisco. In 2022 there were 204 weekly flights heading to New York from San Francisco [45]. This does not take into consideration the number of people on each flight, so human travel patterns are far outranking Kaggle competitions if the lower bound value is correct.

With the lack of certainty, more should be done to make an evaluation feasible and prioritize a focus on carbon footprint. There should be as much transparency as possible regarding the competitions. For example, all teams should submit publicly available code for evaluation. Each team should also be required to submit certain statistics, such

as hardware used, total runtime, and the number of times they trained. Furthermore, teams could be encouraged to use tools such as CarbonTracker, and report both their CO_2 emissions from training their final model as well as the cumulative CO_2 emissions of all training sessions during development. Furthermore, this cumulative carbon footprint can be incorporated as a metric to choose the winner. This would encourage participants to be more carbon conscious.

B. Competition Format Increases Carbon Footprint While Causing Diminishing Returns

Research by Varoquaux et al. show that a focus on outperforming other algorithms leads to *diminishing returns* [5]. Outperforming other algorithms is the basis for a competition, especially if there is a reward. In the study, they investigated Kaggle challenges and found six out of eight of the competitions to have diminishing returns, where the gains made by the top 10% of methods are smaller than the expected noise when evaluating a method. This means that with a different split in the data in the public and private test sets the winner will be different, and the winner does not necessarily indicate a better model.

The competition format for ML causing diminishing returns does not seem to change depending on the level of reward. In other words, a high rewards does not necessarily equate to algorithmic improvements of the winner. Out of the eight studied, the two where the winner showed algorithmic improvements did not have the highest reward. In the SIIM-FISABIO-RSNA COVID-19 challenge, the winner gap was larger than the evaluation noise, meaning that the winning method made substantial improvements compared to the 10% competitor [5]. However the reward was 100,000 USD. The highest reward out of those competitions was 1,000,000 USD from the Lung Cancer Classification challenge, which was shown to have diminishing returns.

While experiencing diminishing returns, the challenge format seems to encourage high activity which implies a large carbon footprint. The average total submissions for a *code* competition is 24,894. There were 32,307 total submissions for the SIIM-FISABIO-RSNA COVID-19, which falls within the interquartile range within this category (see Fig. 4). While the cost of training a model can vary based on task, dataset, and other factors, the high number of submissions still indicates a high carbon footprint. The competition format therefore causes individual teams to have a narrow focus on outperforming other algorithms while leading to a higher carbon footprint.

Future work could look into whether an increase in reward leads to more submission activity, and therefore a higher carbon footprint, and whether this correlates with diminishing returns.

C. It Is Unclear if the Carbon Cost Is For Nothing

There is significant potential in using the solutions for the medical imaging code competitions for practical use. A literature review in 2020 by Malik et al. [46] found that transfer learning based ML models performed on par with medical experts. There are many studies that speak of the potential for AI in healthcare [47–49], but there exist significant barriers to it being implemented into clinical practice.

It is unclear whether the solutions for this competition have been deployed to clinical practice. SIIM and the other co-hosts have stated their intention to deploy the winning algorithms into clinical use [50]. However, we have been unable to find evidence if they have done so. Further studies can look into whether solutions to Kaggle challenges are being applied to clinical practice, or if the main beneficiaries are the winners of the competition.

D. Proposed Solution to Code Reproducibility

A number of papers have already explored the issues around reproducibility of ML experiments. For example, McDermott et al. [51] consider difficulties particular to healthcare related ML, and different ways in which a study can be considered reproducible. Heil et al. [52] also suggest a range of standards that a paper can be held against, ranging from a minimal bronze standard of paper, code, and models, up to their highest standard where the entire study can be replicated by automatic processes.

We propose a number of solutions to the problems of code reproducibility. First, teams submitting to Kaggle challenges can be encouraged to follow guidelines suggested by any of the already established standards in current research. In addition, when submitting code as “reproducible”, it should come with clear step-by-step instructions. It should be tested to ensure that code runs from scratch on a new system. This would mean starting with a new environment, following the package installation procedure given with the README, and running everything. Ideally, this should be tested by an external person to the project who has never ran the code before. To solve the environment problems, docker containers can be provided with all environment requirements already installed.

E. Kaggle Needs to Find Other Ways To Balance the Competition

The *code* competition format does not, in practice, limit compute. According to Kaggle documentation, a *code* competition is more balanced due to all users having the same hardware allowances [4]. They further state that the winning models tend to be far simpler than the winning models in other competitions due to the compute constraints. However, we have found this to not be true for the top ten solutions

in the SIIM-FISABIO-RSNA COVID-19 challenge. Not only were the solutions not simple, but they also far surpassed the time limit and hardware constraints. If intending to impose compute constraints for the purpose of balancing the playing field, then Kaggle needs to find other ways to do so.

VII. Conclusions

There is a rising carbon footprint of machine learning with medical imaging due to high compute requirements. This seems to be exacerbated by the competition format, which encourages the training of many models without consideration of carbon footprint. In this paper, we have investigated the important issue of the carbon footprint associated with Kaggle medical image competitions. These competitions attract a high number of submissions and consequently, a high number of ML models being trained, often with little performance difference between them.

To evaluate the competition we recreated the training of some medical image analysis models, and used the CarbonTracker tool to measure their carbon cost. We estimated a range using the models with the lowest and highest carbon costs, and found the upper bound could be extremely high, with a carbon footprint equivalent to sending multiple people to space. This research highlights the importance of considering these costs, and prioritizing a focus on carbon footprint. We suggest to standardise the reporting of certain statistics, such as hardware used, total runtime, number of times training a model, and overall team CO_2 emissions. Ideally, further steps can be done to reduce carbon cost, such as using a submissions' CO_2 metric as an additional evaluation of performance in order to encourage this as a priority.

We also discuss other challenges found. For example, the difficulties in reproducing competition submissions, how that affects deployability, and steps that can be taken to make code reproducible. Another challenge is the misalignment between the premise of a *code* competition and the actuality. Changes need to happen to the competition format that allows for the reproducibility of submissions to optimize deployability, and also realign the format to level the playing field between competitors.

There is a lot more work to be done in this area to fully explore these issues and there are some important limitations to this paper. Further work could expand on this study to cover more competitions and evaluate more submissions within a competition. This would refine the cost estimations, allowing them to be more generalizable while emphasizing the true carbon footprint of Kaggle machine learning competitions.

Data Availability

For reproducibility, all data used in our analyses are available on https://github.com/carbonCostKaggle

Code Availability

The GitHub organization contains all the code and documentation necessary to reproduce the results of this project: https://github.com/carbonCostKaggle

References

[1] M.-W. Dictionary, *Carbon footprint*. [Online]. Available: https://www.merriam-webster.com/dictionary/carbon%5C%20footprint.

[2] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[3] L. Chancel, T. Piketty, E. Saez, and G. Zucman, *World Inequality Report 2022*. Harvard University Press, 2022, ch. 6.

[4] Kaggle, *Kaggle home page*. [Online]. Available: https://www.kaggle.com/.

[5] G. Varoquaux and V. Cheplygina, "Machine learning for medical imaging: Methodological failures and recommendations for the future," *NPJ digital medicine*, vol. 5, no. 1, p. 48, 2022.

[6] R. Selvan, N. Bhagwat, L. F. Wolff Anthony, B. Kanding, and E. B. Dam, "Carbon footprint of selecting and training deep learning models for medical image analysis," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part V*, Springer, 2022, pp. 506–516.

[7] EPA, *Greenhouse gas equivalencies calculator*. [Online]. Available: https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator.

[8] IBM, *What is computer vision?* [Online]. Available: https://www.ibm.com/topics/computer-vision.

[9] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual review of biomedical engineering*, vol. 19, pp. 221–248, 2017.

[10] O. Edenhofer, *Climate change 2014: mitigation of climate change*. Cambridge University Press, 2015, vol. 3.

[11] L. Cai, J. Gao, and D. Zhao, "A review of the application of deep learning in medical image classification and segmentation," *Annals of translational medicine*, vol. 8, no. 11, 2020.

[12] M. Burkitt, D. Walker, D. M. Romano, and A. Fazeli, "Constructing complex 3d biological environments from medical imaging using high performance computing," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 3, pp. 643–654, 2012.

[13] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 10 039–10 081, 2020.

[14] T. Dettmers and L. Zettlemoyer, "Sparse networks from scratch: Faster training without losing performance," *ArXiv*, vol. abs/1907.04840, 2019.

[15] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *ArXiv*, vol. abs/1909.11942, 2019.

[16] M. A. Abdou, "Literature review: Efficient deep neural networks techniques for medical image analysis," *Neural Computing and Applications*, vol. 34, no. 8, pp. 5791–5812, 2022.

[17] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the carbon emissions of machine learning," *arXiv preprint arXiv:1910.09700*, 2019.

[18] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," *arXiv preprint arXiv:2007.03051*, 2020.

[19] A. Brock, S. De, S. L. Smith, and K. Simonyan, "High-performance large-scale image recognition without normalization," in *International Conference on Machine Learning*, PMLR, 2021, pp. 1059–1071.

[20] P. Luo, X. Wang, W. Shao, and Z. Peng, "Towards understanding regularization in batch normalization," *arXiv preprint arXiv:1809.00846*, 2018.

[21] J. Loy, *Nfnets explained — deepmind's new state-of-the-art image classifier*. [Online]. Available: https://towardsdatascience.com/nfnets-explained-deepminds-new-state-of-the-art-image-classifier-10430c8599ee.

[22] nvnnghia, *Siim2021*. [Online]. Available: https://github.com/nvnnghia/siim2021.

[23] Andrew Kemp, Anna Zawacki, Chris Carr, George Shih, John Mongan, Julia Elliott, Kaiwen, ParasLakhani, Phil Culliton, *Siim-fisabio-rsna covid-19 detection*, 2021. [Online]. Available: https://kaggle.com/competitions/siim-covid19-detection.

[24] Kaggle, *Siim covid-19: Resized to 512px png*. [Online]. Available: https://www.kaggle.com/datasets/xhlulu/siim-covid19-resized-to-512px-png.

[25] N. I. of Health, *Nih chest x-rays*. [Online]. Available: https://www.kaggle.com/datasets/nih-chest-xrays/data.

[26] M. D. L. I. Vayá *et al.*, "Bimcv covid-19+: A large annotated dataset of rx and ct images from covid-19 patients," *arXiv preprint arXiv:2006.01174*, 2020.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[28] D. Juodelyte, A. J. Sánchez, and V. Cheplygina, "Revisiting hidden representations in transfer learning for medical imaging," *arXiv preprint arXiv:2302.08272*, 2023.

[29] N. Codella *et al.*, "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1902.03368*, 2019.

[30] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018.

[31] W. Al-Dhabyani, M. Gomaa, H. Khaled, and A. Fahmy, "Dataset of breast ultrasound images," *Data in brief*, vol. 28, p. 104 863, 2020.

[32] D. S. Kermany *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[33] N. Bien *et al.*, "Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of mrnet," *PLoS medicine*, vol. 15, no. 11, e1002699, 2018.

[34] L. Pedraza, C. Vargas, F. Narváez, O. Durán, E. Muñoz, and E. Romero, "An open access thyroid ultrasound image database," in *10th International symposium on medical information processing and analysis*, SPIE, vol. 9287, 2015, pp. 188–193.

[35] M. Risdal and T. Bozsolik, *Meta kaggle*, 2022. DOI: 10.34740/KAGGLE/DS/9. [Online]. Available: https://www.kaggle.com/ds/9.

[36] G. Hsueh, "Carbon footprint of machine learning algorithms," 2020.

[37] SLURM, *Slurm overview*. [Online]. Available: https://slurm.schedmd.com/overview.html.

[38] AJ_Buckeye, Jacob Kriss, Josette_BoozAllen, Josh Sullivan, Meghan O'Connell, Nilofer, Will Cukierski, *Data science bowl 2017*, 2017. [Online]. Available: https://kaggle.com/competitions/data-science-bowl-2017.

[39] U. o. M. Center for Sustainable Systems, *Carbon footprint factsheet*, 2018. [Online]. Available: https://bit.ly/2Qbr0w1.

[40] T. Schlossberg, *Flying is bad for the planet. you can help make it better.* [Online]. Available: https://bit.ly/2Hw0xWc.

[41] A.-L. Ligozat and S. Luccioni, "A practical guide to quantifying carbon emissions for machine learning researchers and practitioners," Ph.D. dissertation, MILA; LISN, 2021.

[42] S. An, G. Bhat, S. Gumussoy, and U. Ogras, "Transfer learning for human activity recognition using representational analysis of neural networks," *ACM Transactions on Computing for Healthcare*, vol. 4, no. 1, pp. 1–21, 2023.

[43] Y.-M. Kim, J.-B. Poline, and G. Dumas, "Experimenting with reproducibility: A case study of robustness in bioinformatics," *GigaScience*, vol. 7, no. 7, giy077, 2018.

[44] J. Wen *et al.*, "Convolutional neural networks for classification of alzheimer's disease: Overview and reproducible evaluation," *Medical image analysis*, vol. 63, p. 101 694, 2020.

[45] SFO, *Sfo fact sheet*. [Online]. Available: https://www.flysfo.com/about/about-sfo/sfo-fact-sheet.

[46] H. Malik, M. S. Farooq, A. Khelifi, A. Abid, J. Nasir Qureshi, and M. Hussain, "A comparison of transfer learning performance versus health experts in disease diagnosis from medical imaging," *IEEE Access*, vol. 8, pp. 139 367–139 386, 2020. DOI: 10.1109/ACCESS.2020.3004766.

[47] K.-H. Yu, A. L. Beam, and I. S. Kohane, "Artificial intelligence in healthcare," *Nature biomedical engineering*, vol. 2, no. 10, pp. 719–731, 2018.

[48] M. D. Abràmoff, P. T. Lavin, M. Birch, N. Shah, and J. C. Folk, "Pivotal trial of an autonomous ai-based diagnostic system for detection of diabetic retinopathy in primary care offices," *NPJ digital medicine*, vol. 1, no. 1, p. 39, 2018.

[49] F. De Dombal, D. Leaper, J. R. Staniland, A. McCann, and J. C. Horrocks, "Computer-aided diagnosis of acute abdominal pain," *Br Med J*, vol. 2, no. 5804, pp. 9–13, 1972.

[50] SIIM, *Siim, fisabio, & rsna recognize winners of the covid-19 detection & localization challenge on kaggle*, 2021. [Online]. Available: https://siim.org/news/news.asp?id=580751&terms=%5C%22siim%5C%2c+and+fisabio%5C%2c+and+rsna+and+recognize+and+winners+and+co%5C%22.

[51] M. B. A. McDermott, S. Wang, N. Marinsek, R. Ranganath, L. Foschini, and M. Ghassemi, "Reproducibility in machine learning for health research: Still a ways to go," *Science Translational Medicine*, vol. 13, no. 586, eabb1655, 2021. DOI: 10.1126/scitranslmed.abb1655. eprint: https://www.science.org/doi/pdf/10.1126/scitranslmed.abb1655. [Online]. Available: https://www.science.org/doi/abs/10.1126/scitranslmed.abb1655.

[52] B. J. Heil, M. M. Hoffman, F. Markowetz, S.-I. Lee, C. S. Greene, and S. C. Hicks, "Reproducibility standards for machine learning in the life sciences," *Nature Methods*, vol. 18, no. 10, pp. 1132–1135, 2021.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Veronika Cheplygina for her valuable insights, expertise, and support throughout the project. Her quick discernment skills were instrumental in helping us make sense of complex issues and see value in what initially seemed like setbacks. We greatly appreciate her guidance and mentorship. We would also like to thank Lottie Greenwood for her assistance in resolving the many many challenges we faced while working with the HPC. Her problem-solving skills and dedication made this project possible.

APPENDIX A
COMPARISON DATA FOR FIG. 5

SIIM-FISABIO-RSNA COVID-19 Competition	CO_2 (g)
Lower Bound Cost for 31,751 entries	50,262
Upper Bound Cost for 31,751 entries	182,656,518

Familiar Consumption	CO_2 (g)
Air Travel, 1 Passenger, NY $\leftrightarrow$ SF	899,927
Human Life, avg, 1 Year	4,999,948
American Life, avg, 1 Year	16,400,086
Car, Avg Incl. Fuel, 1 Lifetime	57,152,638
Space Flight, 11 Minute, 1 person	75,000,000

TABLE V: Estimated total CO_2 emissions from the SIIM Kaggle Competition, compared to familiar consumption [3, 39, 40].