Leveraging Computer Vision to Map Cell Tower Locations to Enhance School Connectivity

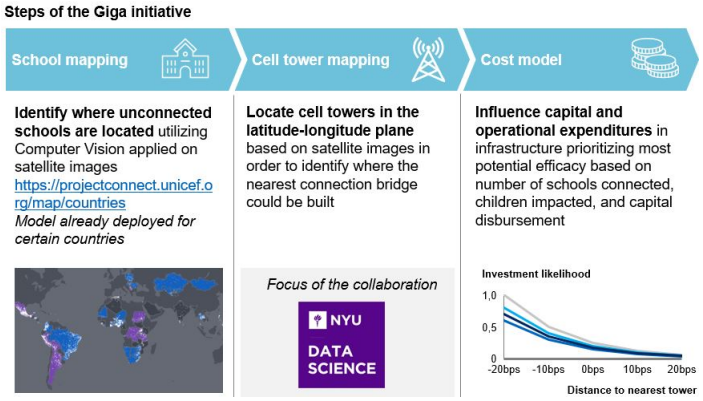
Lorena Piedras **Priya Dhond** **Alejandro Sáez**
Center for Data Science
New York University
lp2535@nyu.edu pnd220@nyu.edu as15796@nyu.edu
Group 3, Project 7
Project Mentor: Mark Ho

1 Introduction

With 4.9Bn internet users, the world is becoming increasingly connected. This widespread network accessibility enables, among other things, increased developmental opportunities and a democratized platform through which the equity gap can be bridged. However, there still are 1.3Bn unconnected children worldwide UNICEF [2020]. The lack of internet connectivity prevents them from accomplishing their educational needs and exacerbates existing inequalities.

UNICEF is an agency of the United Nations responsible for providing humanitarian and developmental aid to children internationally. Two of the pillar values in their mission statement are to *"Advocate for the protection of children's rights"* and *"Ensure special protection for the most disadvantaged children"*.

In this context, UNICEF together with the International Telecommunications Union (ITU) launched in 2019 a specialized global plan, the Giga initiative, with the goal to connect every school to the Internet by 2030. This initiative comprises three key efforts portrayed below.

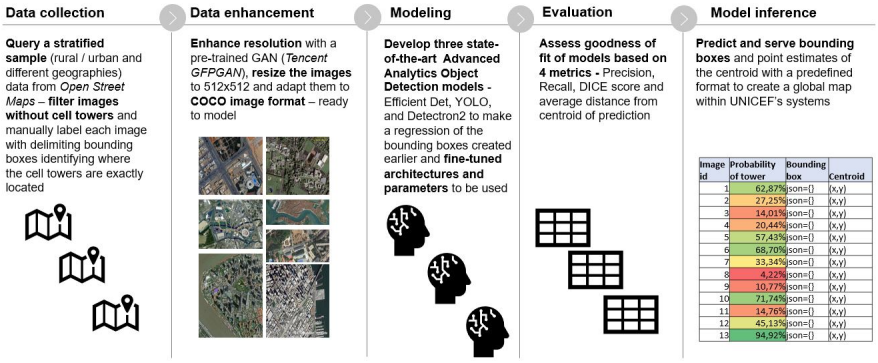


In this paper, and more broadly in the NYU/CDS-UNICEF collaboration, our aim is to support the Giga initiative by building Advanced Analytics models capable of geo-referencing cell towers given Satellite images. The outputs of our models, together with the geo-referencing of unconnected schools, and a cost model will influence private and public capital expenditures in the most cost effective manner, i.e. positively impacting the most number of children given some budgetary constraints.

The collaboration described in this paper has been an effort comprised from September to December of 2022 and conducted by seven team members - Dohyung Kim (UNICEF Project Manager), Iyke Maduako (UNICEF Lead Researcher), Aye Nyein Thaw (UNICEF Research Analyst), Mark Ho

(NYU Faculty Mentor), Lorena Piedras (CDS student), Priya Dhond (CDS student), and Alejandro Saez (CDS student).

In order to map cell towers, and successfully conduct the collaboration, our approach is composed of 5 sequential steps. First, we gathered open-source images and labelled them manually. Then, we enhanced the initial resolution by applying a pre-trained GAN. It's important to note that we didn't test training the model with non-enhanced images so we don't have a clear understanding on the impact of enhancing the images, apart from positive results from third-party literature Shermeyer and Van Etten [2019]. As a third step, we trained three state-of-the-art object detection models to regress the bounding boxes created earlier. After building the models, we evaluated their goodness of fit across a grid of metrics and selected the best performing one (YOLO). Finally, we built an end-to-end pipeline capable of going from raw data collection to prediction serving and mapped the entirety of Brazil with predicted cell tower locations. An illustrative exhibit of this approach can be seen below.



In the following subsections of this paper we will cover related work in this field, the problem definition and algorithms used from a formal and theoretical perspective, the evaluation of our effort, and closing remarks on conclusions and lessons learned through this process.

2 Related Work

Object detection is a complex computer vision problem with a myriad of applications in the medical field, self-driving cars, identity detection and more. It involves classification and location of objects in a image or video Shermeyer and Van Etten [2019] .There has been extensive research on the development of object detection models. Initial approaches used ensembles of hand-crafted feature extractors such as Viola-Jones Detection algorithm Viola and Jones [2001]. The introduction of neural networks such as AlexNet Krizhevsky et al. [2017] considerably increased the accuracy for the task. The current state of the art models are one-stage object detection neural networks such as YOLO Redmon et al. [2016], and EfficientDet Tan et al. [2020a], and transformer models such as Facebook's DeTR Carion et al. [2020].

Detecting objects in satellite images is a challenging task with a small amount of labelled data. In contrast to other disciplines there exists few training datasets that contain satellite images. Additional to lack of data, the task is complicated because satellite images can have very dense backgrounds, objects of varying sizes and often the object of interest has small spatial extent and is difficult to detect. Augmenting the images as a preprocessing step can help increase accuracy, Shermeyer and Van Etten [2019] show impressive results with an increase of 13-36% on mAP after using Very Deep Super-Resolution (VDSR) and Random-Forest Super-Resolution (RFSR).

3 Problem Definition and Algorithms

3.1 Task

Cell tower locations are a critical piece of information for the Giga Initiative. With this knowledge, UNICEF leaders and government officials can use geographical locations of towers, compared to locations of surrounding schools, to gain insight on internet school connectivity. With this

information, individuals involved with the Giga initiative can work with governments to renegotiate internet contracts, provide critical location data to other partners (such as those involved in emergency situation response), and ultimately work with connectivity partners to connect schools and empower students.

While there is no concrete database of cell tower locations, UNICEF has the ability to query satellite images. Thus, our task is to leverage these satellite images to create an object detection model. In the future, this model will be deployed to images in various countries to obtain concrete, geographically-accurate information about locations of cell towers across the globe.

The inputs for the object detection model that we developed are 512×512 pixel images of cell towers and their surroundings and corresponding "bounding boxes". These bounding boxes, individually-created by our team, capture information about each cell tower, corresponding shadow, and corresponding tower base. After developing and training our model, the model will detect the presence of cell towers in unseen satellite images and output bounding boxes and related confidence scores for detected cell towers.

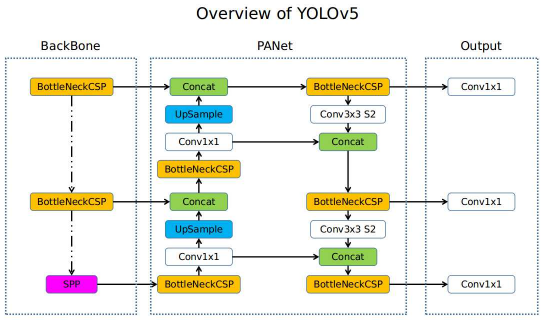
3.2 Algorithms

For this endeavor, we used three pre-trained object detection models adapted to our particular task - **YOLOv5**, **EfficientDet**, and **Detectron2**. While there are a many pre-trained object detection models, we chose these three models for a few different reasons. First, these were used by the UNICEF team in a previous object detection model that identified school locations. YOLO models are typically the standard for object detection, due to their speed and accuracy. However, certain versions of EfficientDet have been shown to perform better than YOLO Tan et al. [2020b]. In addition, in previous UNICEF projects, the use of Fast R-CNN has been successful- leading us to choose to experiment with Detectron2 as well. After implementing these 3 pre-trained object detection models, we compared the image results and metrics produced by each model and proceeded to use one model for an inferencing pipeline. Below we present an brief introduction of the aforementioned models together with links for further reference.

3.2.1 YOLOv5

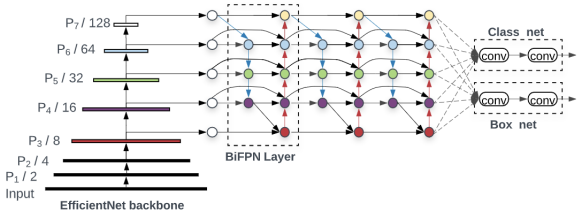
Before YOLO, object detection models treated the task as a two step process: in the first pass the model would present candidate regions where the object could be located and in the second pass the model would classify the object and adjusts bounding box coordinates. The name YOLO stands for You Only Look Once and as the name implies YOLO models are single stage detectors, reframing the problem as a regression. The model separates the image into a N$\times$N grid and predicts bounding box coordinates, confidence scores and conditional class probabilities Redmon et al. [2016]. The network architecture for YOLO is inspired on GoogLeNet model, containing convolutional and fully connected layers.

There have been many changes and developments in the YOLO model since it was first launched, we are currently in YOLOv7. Recent advances have focused on optimizing its architecture and training times while increasing accuracy. A lot of object detection applications such as multi-object tracking and self-driving cars require real-time online predictions, so having efficient models is paramount Wang et al. [2022]. We're using YOLOv5, it was the best performing model in the school image detection project and we wanted to leverage the infrastructure already built to put the model into production. An exhibit of the architecture used is portrayed below:



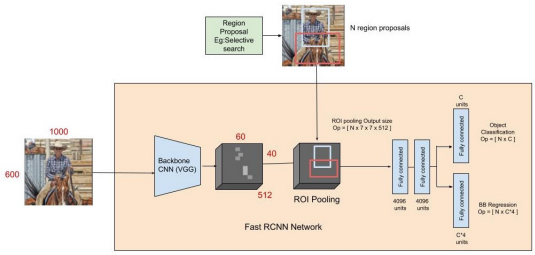
3.2.2 EfficientDet

EfficientDet, a family of object detection models created by Google, are comprised of an EfficientNet backbone and bi-directional feature network (BiFPN) Tan et al. [2020b]. There are 8 different versions of EfficientDet: EfficientDet-D0 to EfficientDet-D7. Each of these versions varies slightly in architecture, with the higher numbered versions (EfficientDet-D5 to EfficientDet-D7) producing more accurate predictions and being better suited for high-resolution images. The difference between each EfficientDet pre-trained model stems from the scaling of the EfficientNet backbone, BiFPN and box/class prediction network. The convolutional neural network used as the backbone for EfficientNet-D0 has a particular depth, width, and resolution. The depth, width, and resolution of this network is uniformly scaled (using "compound scaling"), which results in different EfficientNet backbones. Then, each scaled EfficientNet backbone is combined with a jointly scaled BiFPN network and box/class prediction network to create a different EfficientDet model Solawetz [2020]. For this task, we used the EfficientDet-D5 model with weights generated from training on the official COCO 2017 dataset. An exhibit of EfficientDet's architectures can be found below.



3.2.3 Detectron2

Detectron2 is the third model we have leveraged to forecast the delimiting bounding boxes of the manually labelled cell towers. Detectron2 is Facebook's Artificial Intelligence Research's (FAIR) next-generation platform for object detection and segmentation. It was designed for rapid implementation of state-of-the-art architectures and it enables the end-user to try different models within such as Mask R-CNN, RetinaNet, Faster R-CNN, RPN, Fast R-CNN, TensorMask, PointRend, or DensePose. Due to the past success of Fast R-CNN within UNICEF we decided to use this architecture. For further reference on this architecture see Fast R-CNN and the architecture from below.



3.3 Inference Pipeline

After selecting our final model we built an inference pipeline that takes as an input raw images, uses a GAN to enhance them, resizes images to the correct 512×512 size, loads the model's pretrained weights and predicts bounding boxes, and finally georreferences the cell towers into a geographic coordinate system.

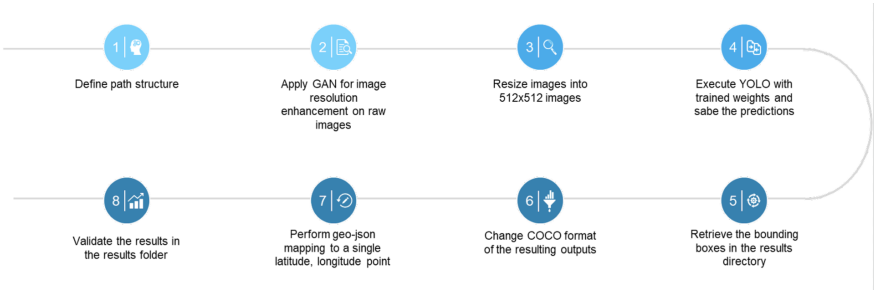


Figure 1: Inference pipeline, going from raw images to geo-referenced cell-towers.

4 Experimental Evaluation

4.1 Data

The data used throughout our project has comprised two main blocks: high-resolution, 512×512 pixel satellite images and COCO JSON annotation files with bounding box coordinates for each image. To obtain the satellite images required for our object detection models, we went through a sequential list of pre-processing steps.

1. Queried the open-source **Open Street Maps** database to collect a list of latitude and longitude coordinates of cell towers in several countries. In line with the priorities of our UNICEF partners, we focused our attention on cell towers in the following countries: Algeria, Argentina, Angola, Brazil, Chad, China, India, Iran, Libya, Mali, Niger, Republic of Congo, Sudan, South Africa, and Saudi Arabia. To ensure robustness of our model in predicting cell towers in a variety of settings, we collected coordinates from both rural and urban settings. We defined "rural" images as images that contained cell towers with very few or no other structures in the image. Typically, these were images of single cell towers against plain or desert backgrounds. In contrast, "urban" images were those with other structures (such as buildings, telephone poles and streetlights) in the background.
2. Verified the presence of cell towers at each of the latitude and longitude coordinates through manual inspection on **Google Maps**. Then, we provided 3000 filtered coordinates (for train/val/test) together with 4000 unfiltered ones (to assess behavior in new images of unknown nature) to our UNICEF mentors who then shared with us satellite images for these coordinates.
3. Enhanced the resolution of the raw satellite images. Initially, the satellite images provided were blurry, which would make detection of cell towers difficult. Thus, we enhanced the images with the **Tencent GFPGAN** and resized the images to 512×512 pixels.
4. Examined each image to visually confirm the presence of a cell tower. Then, using **LabelBox AI**, we created a single bounding box around the tower, shadow, and base in each of the images used for the model. We exported these bounding box annotations as COCO JSON files to be used in conjunction with the satellite images in our pre-trained models.

4.2 Methodology

As described in section 3.2, we tested three different object detection models. In order to select the best performing one, we split our labelled data into tree groups: train, validation and test. Detecting

objects in urban settings is more challenging than rural settings, urban areas typically have very dense backgrounds containing multiple objects that cast shadows such as buildings and light posts. Consequently, keeping the same distribution of urban/rural images within each group was important, so we used stratified sampling. We used a 80% / 10% / 10% (train / val / test) split resulting in 2405, 208 and 301 images for train, validation and test respectively. Finally, we used the train set to further train our pretrained model, the validation set for hyperparameter tuning and assessing model performance and the test set to evaluate the model on unseen data. We used performance on the test set to select our final model, measuring the model's ability to generalize was important because the object detection model will be used for world-wide inference on unseen satellite images.

We tested different metrics to evaluate model results and ultimately employed test precision and recall with IoU of 0.5 to select a model. Initially, we measured average precision and recall with IoU from 0.5 to 0.95 and results were not consistent with a manual error analysis we ran where we sampled examples and visually checked predictions. Precision and recall seemed very low in comparison to the good performance we were seeing in our error analysis and we realized that the 0.5-0.95 IoU threshold was too strict. As described in 4.1 we included the cell tower's shadow and base in the bounding box resulting in bounding boxes that were loose. In conclusion, bounding boxes with a low IoU could still include the cell tower. It should be noted that the UNICEF team was more concerned with having high precision vs. recall since it's more costly to over count cell towers and underestimate infrastructure costs vs. under counting.

To further verify the final model's accuracy, we tested it on unfiltered Open Street Map (OSM) cell tower coordinates. Unfiltered observations represent places where OSM reported a cell tower but the cell tower appearance hasn't been visually verified by us. We ran predictions on satellite images that contained unfiltered coordinates and manually checked 100 images to evaluate the precision and recall of the model.

4.3 Results

Figure 2 shows precision, recall, and DICE for our three models using an IoU threshold of 0.5. We added DICE as an additional metric because it is a proxy for model accuracy. YOLO is the model with highest precision, recall, and DICE on the test set. It is important to note that performance drops sharply in the validation and test set for the three models. Object detection is a very data intensive task and we were only able to train with 3,000 images due to time constraints and data labelling being a very time consuming task, leading to some degree of overfitting.

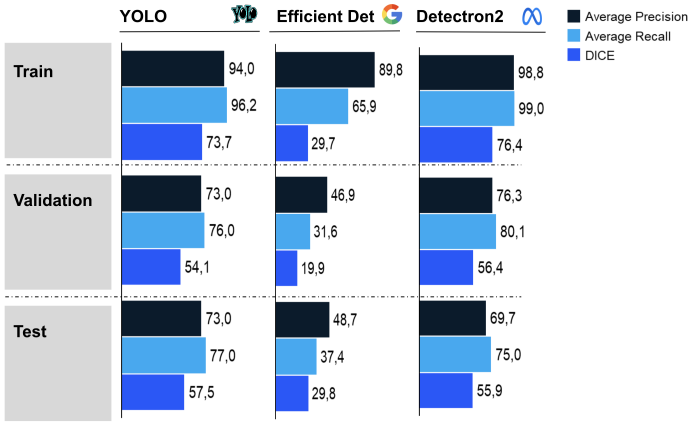


Figure 2: Precision, recall and DICE metrics for our three models using IoU=0.5

In addition to measuring the overall performance, we sampled individual exhibits to get an intuition of why our models were making incorrect predictions in certain instances. In figure 3 we present bounding box predictions for five images in our test dataset. Notably, Detectron2 is able to correctly

detect all cell towers in the five images, YOLO misses the cell tower in image 33 and Efficient Det's bounding boxes do contain the cell towers but are very loose.

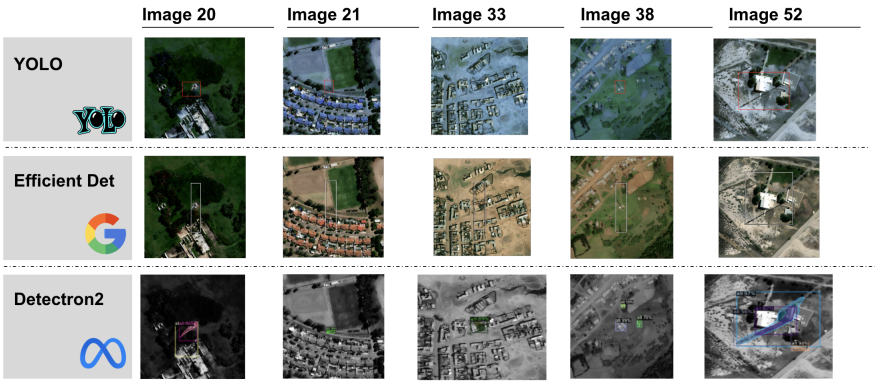


Figure 3: Bounding box predictions for 5 test images

To enrich the previous analysis, we ran the YOLO model on unfiltered OSM data. Precision and recall values are not comparable to those reported in Figure 2. We don't have a bounding box target in this task, so precision and recall are not based on IoU values, to calculate both metrics we visually checked whether there was a cell tower within each predicted bounding box and marked cell towers found in the image but not detected by the model as well. Recall values are high for both rural and urban areas (Figure 4).

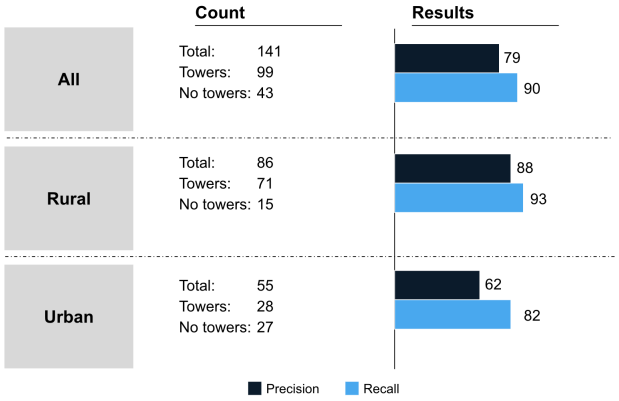


Figure 4: Precision and recall for YOLO's predictions on unfiltered OSM data

We might be overestimating recall since we only accounted for cell towers that were visible from the distance and did not zoom through the entire image. Precision drops sharply in urban settings since they have very dense backgrounds and objects are difficult to detect. Additionally, we have more intra-class variation because shadows change of shape when they are projected over non-flat surfaces such as other buildings or houses. In Figure 5 we find examples that were correctly and incorrectly classified. In rural settings, the model sometimes confuses houses or white objects in a field for cell towers. This could be related to the decision to include cell tower bases on the bounding box of training images. In urban spaces the model confuses cell towers for other objects that cast long and thin shadows such as fences and light posts.

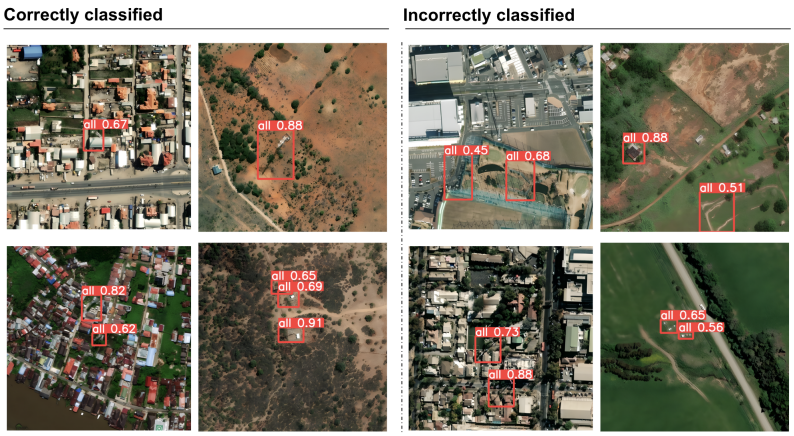


Figure 5: Unfiltered observations predicted correctly and incorrectly by YOLO

In our final prediction step, we geo-referenced the bounding boxes to a geographic coordinate system and used the centroid of each bounding box as a prediction for the cell tower location. We calculated the geodesic distance between the cell tower coordinates and our centroid prediction for observations in the validation and test set. Figure 6 presents a histogram with the distances for the test and validation set, the median distance is at 19 and 17 meters respectively. This result is well within the bounds of what the UNICEF team was expecting.

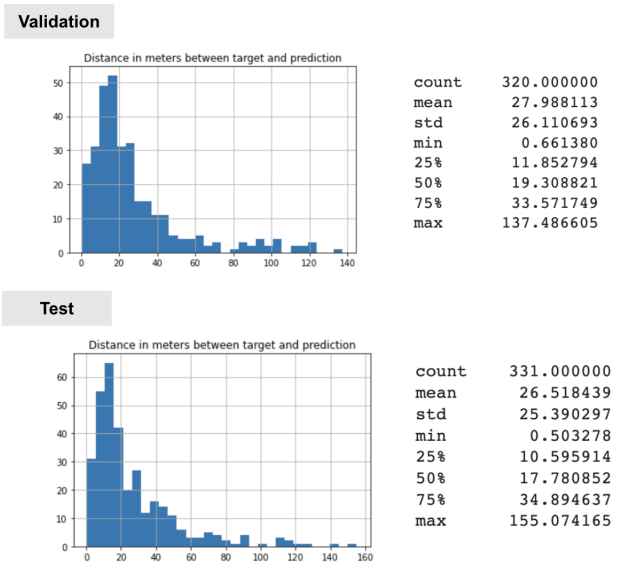


Figure 6: Geodesic distance in meters between cell tower coordinates and YOLO centroid predictions for validation and test set

4.4 Discussion

Having seen the details presented in 4.3, namely the overall model performances in a curated dataset, individual exhibits, and the performance of the model in unseen images, we can derive the following insights.

- **The hypothesis that cell towers can be identified from satellite imagery in an automated fashion is confirmed** by observing high levels of precision in recall across the train, validation, and test sets.
- **Among the three developed models the top performing one is YOLO** with test precisions and recall of 73 and 77 respectively.
- **The performance of our model is greater in rural areas than in urban ones (93 vs 82 precision)**. This is not only to be expected (due to the lack of neighboring buildings) but also desirable, since UNICEF is focusing on rural areas to develop new infrastructure.
- Beyond standardized model performance metrics such as precision, recall, and DICE, **the selected model is capable of regressing where towers are located with high level of proximity (average of 26 meters)**.

For all, our team together with UNICEF leadership decided to move forward with the YOLO model developed and the inference pipeline discussed above. This model has been used to map the entirety of Brazil with two purposes - evaluate the working functionality of the pipeline, and assess the goodness of fit and reliability of the model in unseen large-scale data.

5 Conclusions

As potential next steps discussed during the collaboration, some of which will be made from the UNICEF side, are:

- **Follow a data-centric approach disregarding confusing images**: Among the images with which the model was trained, some induced confusion which could make even a human classifier make mistakes. A hypothesis is that eliminating these images from the train-set could yield higher metrics. This is an experiment that was not conducted due to the already limited size of the data and the limited time.
- **Fine-tune threshold cutoff decision**: In the work discussed above, we went from raw satellite images to a bounding box regression at-scale. However, the end-user, i.e. Giga stakeholders, will need to choose a level of model-confidence which makes business sense. That is, when mapping the entirety of Brazil at a 0.5 cutoff, many towers are predicted which might not be realistic. Finding the optimal cutoff (tradeoff between false posities and false negatives) is something that we did not explore but would come as a natural next step and enhancement to the work discussed here.

6 Lessons Learned

There were several challenges that we came across in applying our data science skill set to this task. Many of our challenges were related to the data used for the project. Since there is no accurate database of location coordinates for cell towers across the globe, after manually querying data points from Open Street Maps, we had to individually verify that each coordinate was a cell tower. To do this, we each verified 1000 coordinates over 3 days, resulting in 3000 data points. Because this was a very manual process, we were not able to verify more coordinates, and thus our dataset throughout the project was very small. Furthermore, the cell towers in our data set are of 3 different types: tall, thin towers, triangular-shaped towers, and wider, cylindrical towers. Unfortunately, variation in towers along with the small number of images resulted in a model with difficulty in confidently identifying towers in the test set of images. After receiving the satellite images of these coordinates, we had to manually draw bounding boxes for each image. This was a very time-intensive process. While we could not overcome this challenge, there was a positive result from this. Since we had manually verified towers and bounding boxes in each image, we were training our models with images that we were confident had a cell tower. Our model would have had poor results if we had used images during training that we had not individually verified for presence of cell towers.

One of the key takeaways from this experience is that volume of data is critical for creating robust object detection models. In further analyzing the results of our object detection model, we realized that some of the poor performance could be attributed to training images that had blurry towers and varying shadows. From this, we learned that data and image quality is key in object

detection. Finally, perhaps the most important takeaway from this project, is that there is never a clear-cut "perfect" object detection model for any given situation. After experimenting with YOLO, EfficientDet, and Detectron2 during the modeling phase of our project, we had to compare the output and results from each model to understand which performed better for our purpose. While we chose to move forward with the YOLO model in this project, this is not to say that YOLO is the best model for any object detection task. Through this, we learned that it important to test different models to identify the best model for a particular business question and purpose. This is certainly a valuable lesson that we will carry into any data science project or task that we encounter in the future.

7 Student Contributions

The NYU-UNICEF collaboration was a truly joint effort. The entire team was in charge of verifying cell tower coordinates, and manually generating bounding boxes. Once the data preparation phase was over, each teammate worked with a different model, Priya was in charge of training and doing error analysis with EfficientDet, Alejandro with Detectron2, and Lorena with YOLOv5. Similarly, during weekly checkpoints the workload of presenting and interacting with the UNICEF team was distributed uniformly. Finally, the team collaborated on writing the report and creating the poster.

As a closing remark, we would like to place on record our deepest sense of gratitude towards the UNICEF and NYU mentors for giving us such an enriching opportunity. This project not only sharpened our Computer Vision skillset but also allowed us to contribute to such a pressing and impactful initiative.

References

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 213–229, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58451-1. doi: 10.1007/978-3-030-58452-8_13. URL https://doi.org/10.1007/978-3-030-58452-8_13.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL https://doi.org/10.1145/3065386.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.91. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91.

Jacob Shermeyer and Adam Van Etten. The effects of super-resolution on object detection performance in satellite imagery. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1432–1441, 2019. doi: 10.1109/CVPRW.2019.00184.

Jacob Solawetz. A thorough breakdown of efficientdet for object detection, 2020. URL https://towardsdatascience.com/a-thorough-breakdown-of-efficientdet-for-object-detection-dc6a15788b73.

M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10778–10787, Los Alamitos, CA, USA, jun 2020a. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.01079. URL https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.01079.

Mingxing Tan, Ruoming Pang, and Quoc Le. Efficientdet: Scalable and efficient object detection, 2020b. URL https://doi.org/10.48550/arXiv.1911.09070.

UNICEF. United nations children's fund and international telecommunication union, "how many children and young people have internet access at home? estimating digital connectivity during the covid-19 pandemic.", 2020. URL https://www.unicef.org/media/88381/file/How-many-children-and-young-people-have-internet-access-at-h.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. doi: 10.1109/CVPR.2001.990517.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022. URL https://arxiv.org/abs/2207.02696.