Lacuna Solar Survey Challenge: Counting Photovoltaic and Solar Panels from Aerial Imagery

Hugo Veríssimo
Complements of Machine Learning 24/25
University of Aveiro
Aveiro, Portugal
hugoverissimo@ua.pt

João Cardoso
Complements of Machine Learning 24/25
University of Aveiro
Aveiro, Portugal
joaopcardoso@ua.pt

***Abstract*—abstact**

Keywords: R, Object Detection

I. Introduction

Access to electricity and warm water has been a basic necessity for a long time in the developed world. In developing countries, the lack of a centralized distribution system makes this access harder for everyone. To improve on this gap, it is essential for governments, non-governmental organizations, and energy suppliers to understand how solar photovoltaic and solar thermal panels (for electricity and water heating, respectively) are distributed throughout the territory to ensure proper planning and effective policy making. In this context, a challenge was created by the Lacuna Fund and associate entities in the Zindi platform to develop a machine learning model capable of accurately detecting and counting the number of solar thermal (solar from here on) and photovoltaic panels in drone and satellite imagery.

In the present work, we have developed different approaches to the problem at hand, where we need to count the two types of panels separately: taking a regression type approach, where images are analysed as a whole and the target number of panels are provided per image; by identifying the panels through object detection and counting them afterwards; and by applying segmentation models to identify the regions of interest, analysing them and counting the panels. The different approaches were developed so that a single model for each type would be able to tackle the task of identifying and counting the different panels.

II. State of The Art

The task of detection of solar panels serves many purposes, with the strongest focus of the works here documented being on the detection and localization of solar panels in large areas (countries). This type of assessment allows for proper planning of infrastructure, and to adapt current maintenance through peak output of dense areas with solar panels.

In the work of Malof *et al.* the authors developed an automatic detection model of solar arrays, specific for photovoltaic panels. The model developed had computational efficiency at the forefront, with the purpose of being deployed nation wide in the United States, hence a multi-stage approach consisting of: pixel-wise feature extraction; Random Forest (RF) Classifier; post-processing to improve the pixel-wise classification accuracy for the object detection phase; finalizing with the object detection, via thresholding the finalized confidence map. This algorithm presents superior performance to standalone RF, and was well remarked as an initial assessment methodology for large areas.

The *DeepSolar* model consists allowed to create a nearly complete contiguous solar panel installation map, and consists of two parts, with different purposes: firstly, via transfer learning the convolutional neural network (CNN) classifier is trained on labeled imagery that merely indicates the presence or absence of panels (though the volume of data is considerable, at almost 400 000 images just for this task). The model is then enhanced by adding an additional CNN branch directly connected to the intermediate layers to add segmentation capabilities to the model. The difference is that this approach allows the model to be trained on the same dataset, to ”greedily” extract the relevant features associated with solar panels, learning in a semi-supervised manner. The model was able to achieved the best result of 93.7 % precision and 90.5 % recall in non-residential areas.

Different models have been proposed for panel detection through image segmentation, using Mask R-CNN He, 2020, TernausNet, based on U-Net Kausika 2021, MobileNet with U-Net backbone Wari, 2021, with different approaches to data preparation and application. These models consistently score higher than 90 % in terms of precision, at the cost of higher computation times compared to the examples previously mentioned (also for very different task sizes). A couple notable aspects across publications: the size of datasets (often in the tens of thousands of images), with strong strategies for collecting and/or preprocessing, ensuring that the models have the best possible starting point to learn and develop; and the focus on a specific type of panel or problem topic, not trying to solve multiple problems with a single tool.

For the purpose of this work, we have developed three approaches to the problem, to assess their strengths and merits, which will be detailed in the next subsections.

A. Image-based Regression

Considering that every image is labelled in terms of amount of photovoltaic and thermal solar panels, the conditions for training considering a regression approach are available. For this, deep neural networks (convolutional neural networks, CNN) were considered for their excellent capabilities in extracting spatial features, but each with a distinguishing feature.

ResNet (Residual Networks) was introduced in 2015 by Microsoft, and is notorious for dealing with vanishing gradient problem by skipping connections between neuron layers. This allows the model to grow and have several layers (over 100), making it capable of learning a larger variety of features from the first to the deeper layers, without suffering from performance degradation.

DenseNet follows on the footsteps of ResNet, and was introduced in 2016 by Huang *et al.*, whereby the idea of skipping connections is taken further, and every layer prior to a given layer is connected to it. This avoids the chain multiplication problem (which leads to vanishing gradients), while reusing previously learned features, retaining relevant information throughout the deepness of the CNN. This architecture also has the benefit of using less parameters than CNN in general (and ResNet as well).

EfficientNet (EffNet), the most recent of this group was introduced by Google AI in 2019, and departs from the direct implementation of the aforementioned methods, dealing with vanishing gradients given its intrinsic architecture. By having MB Conv (Mobile Inverted Bottleneck Convolution) as its building block, it allows to extract spatial features by expanding the number of channels to perform depthwise convolution and applying separate filters per channels **(verificar a nomenclatura)**, that is then compressed to the original size of the input. Whilst this happens, the mechanism of squeeze-and-excitation determines the importance of said features, further addressing the vanishing gradient problem. This architecture is then scaled via compound scaling (rather than arbitrarily), whereby the size of the CNN (depth x width and input resolution) is scaled in a balanced manner, with the compound coefficient ϕ being adjusted by grid search.

B. Object Detection

With object detection, the model is trained based on bounding boxes surrounding the subject of interest, in order to be able to identify them in new environments and types of images. With multiple classes, the model discern between labels with class probabilities, estimated from the extracted features of the assessed boxes (using a CNN) throughout the image. One of the most famous and used models is YOLO ("You Only Look Once") introduced in 2015 by Redmon & Fardhi, being a crucial model in fast object detection and localization. In a single stage, the model addresses the localization and detection problems, where a CNN backbone (that has changed with the released versions) is used, with feature fusion layers (akin to the way learned features are reused, involving various algorithms), and an output of the likelihood of a class and the location of the bounding boxes.

C. Instance Segmentation

Building on the concepts presented before on object detection, instance segmentation addresses the problem of identification by incorporating the information of object masks. These are tighter around the object (than bounding boxes), which serve to add a layer to the YOLO algorithm. A global prototype mask (that works with a separate, smaller CNN) is generated for the whole image - this way once the bounding box is determined, mask coefficients are estimated in order to combine the prototype masks into an instance specific mask. This approach avoids per mask full resolution segmentation, keeping the model size small and fast.

III. Methodology

A. Data description (com EDA)

The dataset was provided within the Zindi challenge, consisting of 4419 images (from drone and satellite sources), and per image metadata, consisting of the source (drone or satellite), mask placement, and context of the installation surroundings (e.g., roof, floor, array).**info do pie chart** The proportion between photovoltaic and thermal solar panels is heavily imbalanced, with 95 % of the dataset corresponding to photovoltaic panels. Of all the images, 3312 (75 %) have detailed information on the masks (location, number of panels within, if they are photovoltaic or thermal). The remaining 1107 (25 %) do not have this information, only the number of photovoltaic and thermal panels in the image. Hence, per design of the competition the train/test split is 75/25.

Fig. 1: Images of photovoltaic panels placed on the roof, from drone (left side) and satellite (right side). The difference in resolution between them is evident.

Fig. 2: The different panel placement possibilities (top right is from satellite imagery). Bottom left image shows an image that is atypical for a drone style image.

In Fig. 2 the image origin and placement are displayed, which according to the challenge information were labelled by expert personnel. Where the placement class wasn't conclusive, the images are labelled as "S-unknown" (the remaining examples are self explanatory). Besides the two classes to be identified, the context and origin of the images means a considerable number of combinations for the model to learn, where some of these combinations might (and certainly are), under-represented given the relatively small dataset.

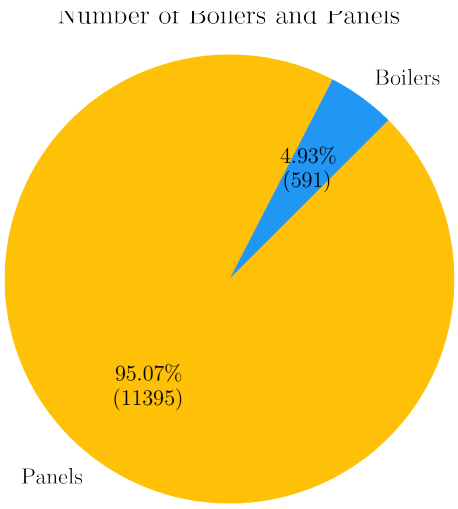
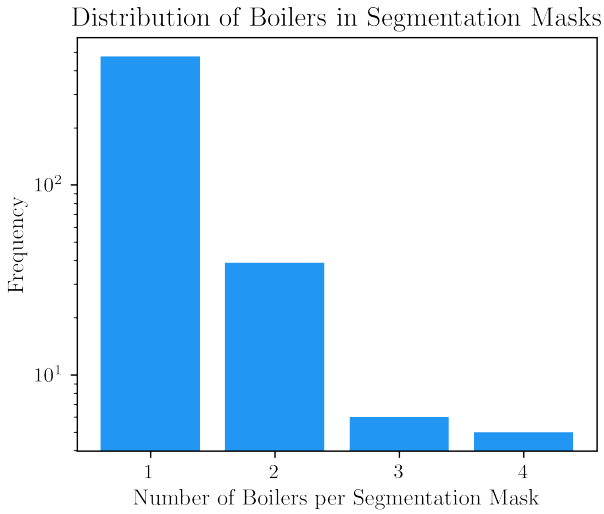
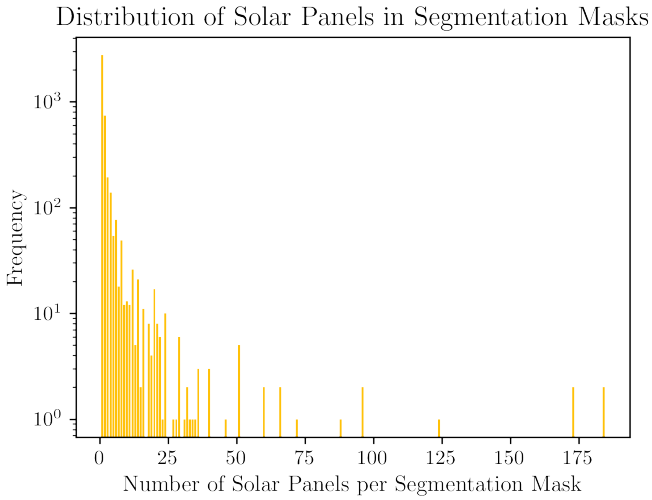


Fig. 3: Portion of the number of photovoltaic and thermal solar panels counted from the available metadata.

The masks are not consistent throughout the dataset, with varying number of panels within them (some contain a single panel, others might contain a complete array of up to 200), which can be clearly seen in 4a



(a) Distribution of the number of thermal solar panels per segmentation mask.



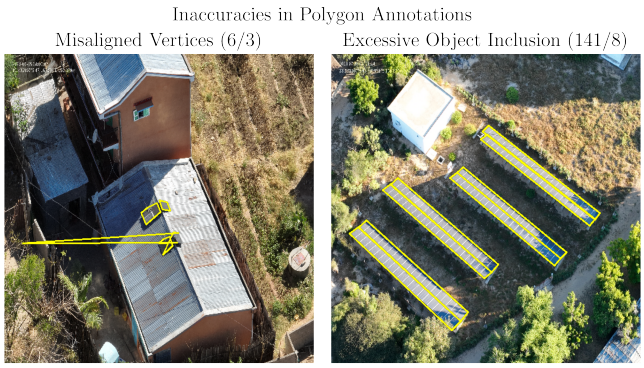
(b) Distribution of the number of photovoltaic panels per segmentation mask.

Fig. 4: Distribution of the number of panels within a single mask.

From the sample images below, the difference in the quality of the masks is stark. Several images were found to have misaligned vertices of the masks, and several masks had a large amount of panels (as seen from the distribution aforementioned). In principle, with the reference of the number of panels within each mask, it might have a small impact for some types of models, but the same model is in fact learning features for different representations: rather than for the representation of individual panels. This was an evident

obstacle to the implementation of YOLO type models, which was dealt with, and further detailed below.

(a) Sample image of accurate labelling, with a single panel per mask.



(b) Sample images of incorrectly marked masks: (on the left) mask distorted, misrepresenting the panel; (on the right) excessive objects within a single mask.

Fig. 5: Sample images from the Zindi dataset.

B. Preprocessing

The discussion forum for the competition was a fruitful source of information on the dataset and how to address it. As seen above, the incorrectly defined masks, and the masks with several panels represented hindrances to achieve the best possible performance. Besides that, some of the masks were shifted from the actual position of the panels, which considering that all were identically shifted, seemed deliberate from the competition. By manually analysing every image it was possible to detect such images (with wrongly drawn masks and shifted), and correct them. Upon completing the dataset revision, 263 of the training samples were lost due to poorly drawn masks (the remaining images that had misaligned masks were corrected).

Once the dataset was ready, the (online) data augmentation process was devised in order to increase the number of training and diversify it, in order to enhance the generalizability of the trained models. The transformations HorizontalFlip, VerticalFlip, RandomRotate90, GaussianBlur, CLAHE (Contrast Limited Adaptive Histogram Equalization), HueSaturationValue and Normalize were applied before all training cycles. The images were all resized to 512x512, to lower the computational burden and homogeneize the code throughout the pipeline.

C. Results

....
aaa

IV. Models

llalal

A. Hybrid Model aka zulo40

tendo em conta a metadata da imagem e a propria, foi criada uma familia de modelos que nao tinham em conta a mask.

esta familia de modelos teve como base o codigo? desenvolvimento por um dos utilizadores da competicao e posteriormente adaptado e melhorado por nos

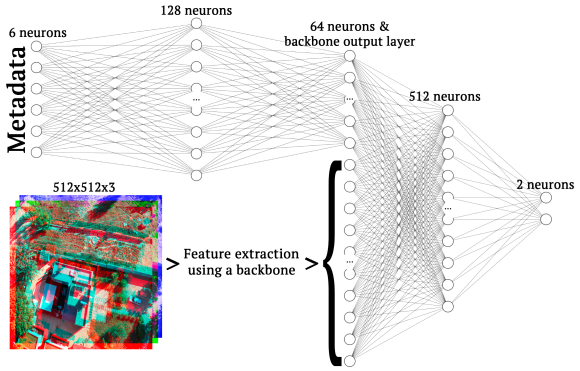


Fig. 6: CAPTION CAPTION CAPTION CAPTION CAPTION

a arquitetura? desta familia de modelos pode ser verificada na imagem 6, que consite num hybrid model de neural networks, ao usar um bacbone (transfer learning) para extrarir features da imagem (cnn's) e snn? para extrar features da metadata. de seguida, é usado um attention head para juntar todas estas features, da imagem e da metadata, para posteriormente obtermos o resultado da regressao

ademais, como backbone foi exprimentado um densenet121, efficientnetv2B3 e resnet101, através da divisao dos dados em treino e validacao e o uso de data augmentation nos dados de treino.

ao verificar o melhor desempenho pelo efnet, tanto no treino como validacao como teste, decisiu-se fazer um fine-tunning dos hyperparameters.

TABLE I: zulo40 type model hyp

Hyperparameter	Possible Values
Batch Size conta?	$\{16, 32, 64\}$
Optimizer	AdamW
Learning Rate	$[10^{-5}, 10^{-3}]$
Weight Decay	$[10^{-5}, 10^{-3}]$
Dropout	$\{0.2, 0.3, 0.4\}$
Scheduler	CosineAnnealingWarmRestarts
T_0	$\{3, 5, 7, 10\}$
T_mult	$\{1, 2, 3, 5\}$
Loss	HuberLoss
δ	1

na tabela III pode se verificar o espaco possivel de hyperparametros utilizado, tendo sido a escolha dos melhores baseada numa random search com o uso de cross-validation.

contudo esta cross validation não foi muito tipica devido aos elevados tempos de ajuste dos modelos. os dados de treino foram separados em 5 folds (80/20), mas cada conjunto de hyperparametros so foi ajustado 3 vezes, e não as tipicas 5 vezes.

de modo a escolher o melhor modelo usou-se o validation mae médio como métrica tendo sido o melhor conjunto de hyperparametros dado por

- batch 16
- leraning 1e-4
- weight dec 1e-4
- dropout 0.4
- t0 5
- tmul 2

meta06

para alem disso é tbm importante indicar o uso de:

- juntar os modelos (3, pq sao 3 folds) e faz a media das previsoes
- Test-Time Augmentation (TTA), where the model makes multiple predictions on augmented versions of the same image (e.g., flipping, scaling, cropping). These predictions are later averaged to improve accuracy.
- o scheduler altera o learning rate e o paraetro de regularizacao

quanto aos resultados deste modelo ...

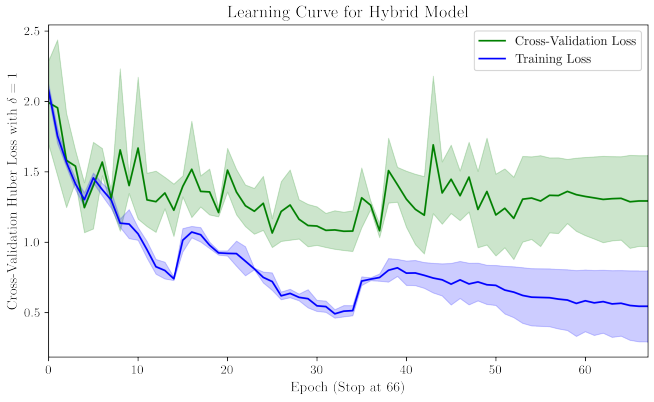


Fig. 7: CAPTION CAPTION CAPTION CAPTION CAPTION

a learning curve Fig. 7.... é de notar q ela revela picos em certar epochs (porximo da 5, 15, 35) devido ao scheduler aumentar o parametro de regularizacao

para alem disso a mesma parou a epoch 66 por early stopping, sendo o criterio de paragem a melhor validation mae (ns se e early stopping pq n tem patience mas sim max de 75 epochs e dps escolher o melhor)

no signs of overffiting

TABLE II: Error metrics for the Hybrid Model, for the train and test set, along with the number of samples for each.

Dataset	MAE	Support
Train Set	0.5127	3312
Test Set	0.8434	1107

quanto as metricas de erro presentes na Table II, apesar de so termos o MAE parece n haver sinais de overiffing pelas medidas serem semelhantes

B. Segmentation Model

isto tbm é do yolo de obj id, ns onde se meta: ideia inicial era fazer segmentacao de conjuntos de paineis e de boilers e posteriormente com outro modelo fazer contagem. problema foi q o modelo ao reconhecer paineis invidivudais, mesmo em conjuntos identificava individuais, levando ao mau desempenho do modelo

entao decidiu-se rever o dataset e separar manualmente os poligonos em paineis e boileres indivudais, deixando de existir grupos de paineis

o yolo apresentou assim mt melhores resultados, apesar da reducao da dimensao do dataset em termos de imagens mas aumento de amostras do que realmente sao paineis e boilers individuais

relativamente ao yolo seg:

de modo a encontrar o melhor modelo de segmentacao dos paineis e boilers, tal como no efnet (depende se vem dps ou antes) os dados foram dividicos em 5 folds, e para cada conjunto de hyperparametros os modelos foram ajustados em 3 conjuntos diferentes de 4 folds sendo validado no restante, e o modelo escolhido foi aaquele com menor mae médio nos folds de validacao

o conjunto de pesquisa de hyperparemetros é dado pela tabela seguinte, mas é de notar q nsao sao testadas todas as combinacoes mas apenas algumas aleatorias

TABLE III: yolo seg model hyp

Hyperparameter	Possible Values
Batch Size conta?	$\{8, 32, 16\}$
yolo	{yolo11l-seg, yolo11m-seg, yolov8l-seg}
imgsize	512
augment	True
patiente	$[10, 25]$
cls	$[0.5, 2.5]$
lr0	$[10^{-4}, 10^{-3}]$
lrf	$[0.01, 0.1, 1]$
mixup	$[0, 0.5]$
copy_paste	$[0, 0.8]$
scale	$[0.5, 1]$

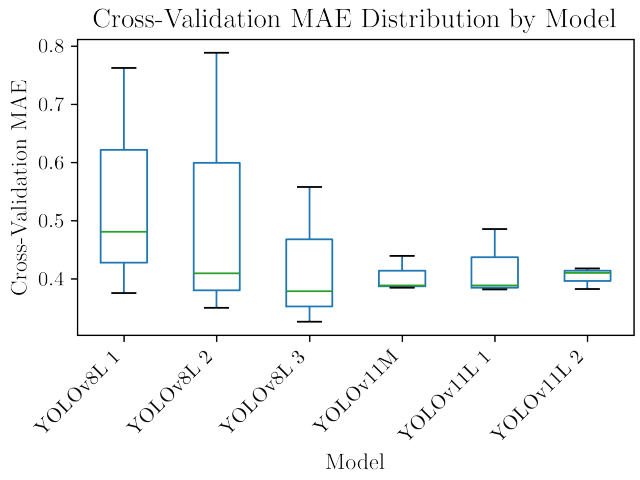


Fig. 8: CAPTION CAPTION CAPTION CAPTION CAPTION (gerado pelo modulo do yolo automaticamente) referir isso pq ele é ”diferente” dos outros ig

a fig 8 demonstra o mae para os diferentes modelos ajustados na busca dos melhores hyperparemtros, tendo sido o YOLOv11L 2 a obter o menos MAE médio nas suas validações, tendo sido ele o escolhido. seguem-se abaixo os seus hyperparemetros

- yolo yolo11l-seg - imgsize 512 - augment True - patiente 25 - lr0 10^{-3} - lrf 0.1 - cls 0.5 - mixup 0 - copy paste 0 - scale 0.5

os restantes parametros sao os que vêm por defeito no package ultralytics referentes ao modelo yolo11l-seg

quanto aos resultados do modelo em referencia, é de notar que o mesmo foi treinado tendo em conta o segmentar os objetos (pan e boilers) da melhor forma possivel, pelo que as metricas do ajuste do modelo sao referentes a isso e nao ao problema de contagem, sendo esta contagem apenas feita apos o modelo ajustar os segmentos as imagens

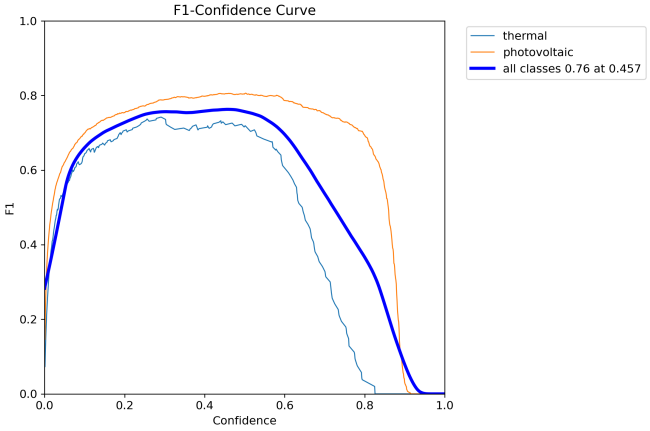


Fig. 9: CAPTION CAPTION CAPTION CAPTION CAPTION (gerado pelo modulo do yolo automaticamente) referir isso pq ele é ”diferente” dos outros ig

a fig. 9 mostra a metrica f1 relativa a segemntacao do segundo fold do modelo, revelando um melhor desempenho dos photovoltaic em relacao aos thermal, mas sem uma diferenca significativa.

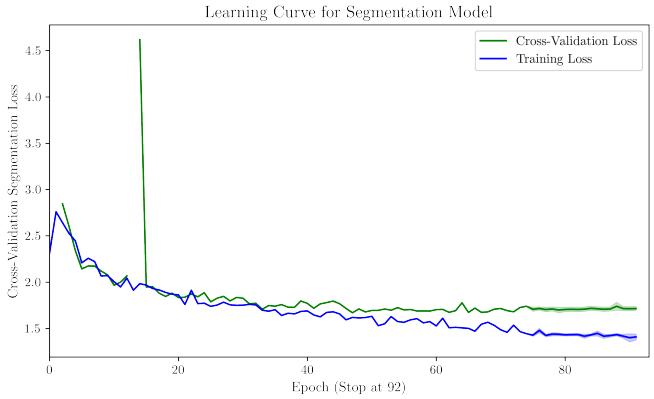


Fig. 10: CAPTION CAPTION CAPTION CAPTION CAPTION

na fig. 10 pode-se verificar a learning curve do melhor modelo, a demonstrar a sua convergencia sem sinais de overiffint. apenas com um possivel outlier na loss, que pode ter devido a algum erro de computacao por parte do calculo do segmentation loss por parte do yolo. é possível ver tbm a paragem do modelo após 92 epochs por early stopping.

TABLE IV: Error metrics for the Segmentation Model, for the train and test set, along with the number of samples for each.

Dataset	MAE	Support
Train Set	1.5645	3312
Test Set	1.3415	1107

quanto as metricas relacionadas com a contagem dos pan e boil, o nosso problema original, o seu resultado é apresentado na tabela IV

V. Discussion

https://zindi.africa/competitions/lacuna-solar-survey-challenge/discussions/25674

A. Performance Metrics

discussao

VI. Conclusion

conc

Work Load

Both authors contributed equally to the project.

References