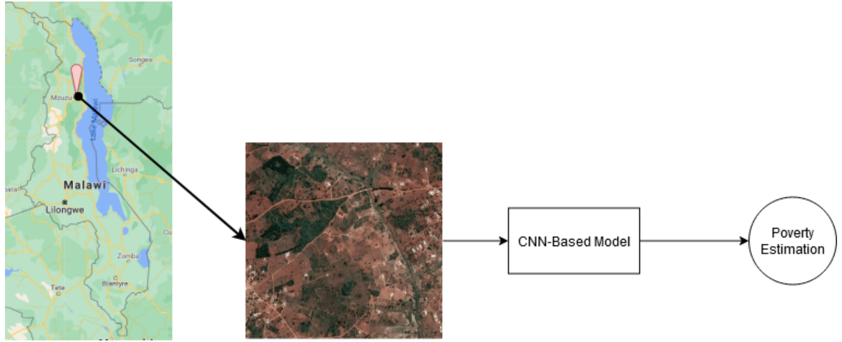
Assessment of CNN-based Methods for Poverty Estimation from Satellite Images

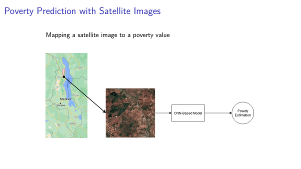
Robin Jarry, LIRMM, CNRS & Univ. Montpellier, France
Marc Chaumont, LIRMM, CNRS & Univ. Montpellier, Univ. Nîmes, France
Laure Berti-Équille, ESPACE-DEV, Univ. Montpellier, IRD, UA, UG, UR, France
Gérard Subsol, LIRMM, CNRS & Univ. Montpellier, France

Hello every one, I'm Robin Jarry. I'm a PhD student at LIRMM, a french computer science laboratory. The work I'm going to talk about was made under the supervision of Marc Chaumont, Laure Berti-Equille and Gerard Subsol, and is titled Assessment of CNN-based methods for poverty estimation from satellite images.

Poverty Prediction with Satellite Images

Mapping a satellite image to a poverty value





Let me introduce the context of poverty prediction with satellite images. [HERE] To create accurate policies in most developping countries, it's important to have reliable poverty estimation, but as I'll explain later, ground truth poverty estimation are scarce in most developping countries. Thus, we are trying to map a satellite image, depicting rural village or larger cities, to a real number that's representing poverty on the image. As it showed great promises, we focus on mehtods that use Covolutional Neural Networks.

[SLIDES]

As a starting point, let's consider a simple methodology where a CNN is trained on pairs of examples of satellite images and ground truth poverty values.

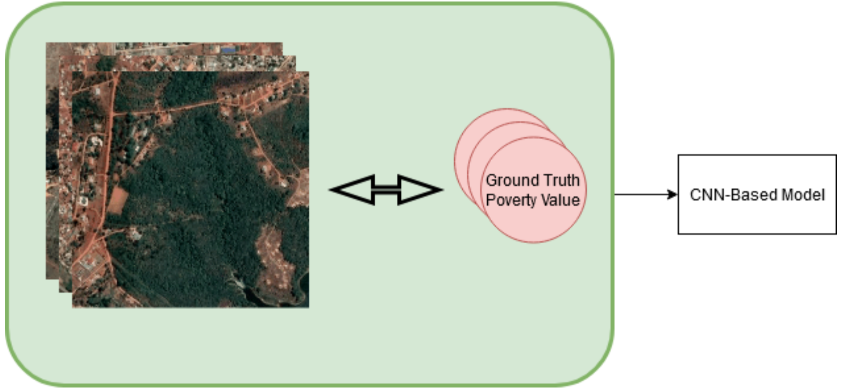
[SLIDES]

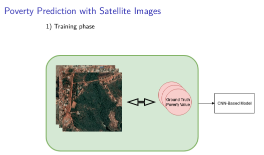
After this training, a CNN model is ready to estimate poverty from any other satellite image.

In fact, this naïve methodology, used in several other image processing tasks can't work here because we are facing two major difficulties.

Poverty Prediction with Satellite Images

1) Training phase



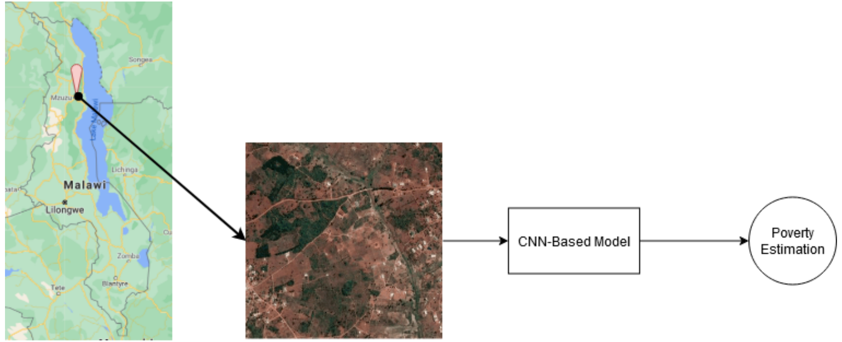


Let me introduce the context of poverty prediction with satellite images. [HERE] To create accurate policies in most developping countries, it's important to have reliable poverty estimation, but as I'll explain later, ground truth poverty estimation are scarce in most developping countries. Thus, we are trying to map a satellite image, depicting rural village or larger cities, to a real number that's representing poverty on the image. As it showed great promises, we focus on mehtods that use Covolutional Neural Networks.
[SLIDES]
As a starting point, let's consider a simple methodology where a CNN is trained on pairs of examples of satellite images and ground truth poverty values.
[SLIDES]
After this training, a CNN model is ready to estimate poverty from any other satellite image.

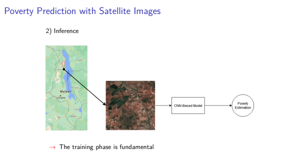
In fact, this naïve methodology, used in several other image processing tasks can't work here because we are facing two major difficulties.

Poverty Prediction with Satellite Images

2) Inference



→ The training phase is fundamental



Let me introduce the context of poverty prediction with satellite images. [HERE] To create accurate policies in most developping countries, it's important to have reliable poverty estimation, but as I'll explain later, ground truth poverty estimation are scarce in most developping countries. Thus, we are trying to map a satellite image, depicting rural village or larger cities, to a real number that's representing poverty on the image. As it showed great promises, we focus on mehtods that use Covolutional Neural Networks.

[SLIDES]

As a starting point, let's consider a simple methodology where a CNN is trained on pairs of examples of satellite images and ground truth poverty values.

[SLIDES]

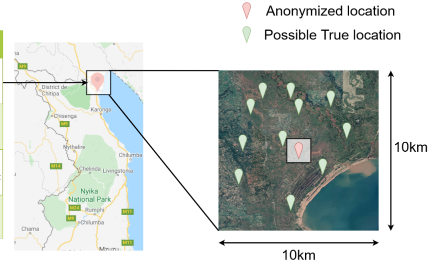
After this training, a CNN model is ready to estimate poverty from any other satellite image.

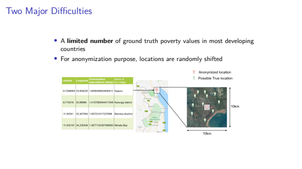
In fact, this naïve methodology, used in several other image processing tasks can't work here because we are facing two major difficulties.

Two Major Difficulties

- A **limited number** of ground truth poverty values in most developing countries
- For anonymization purpose, locations are randomly shifted

Latitude	Longitude	Consumption expenditures ($/day)	Name of the village
-9.7298955	33.859230	1.8056488929595411	Kaporo
-9.715316	33.88666	1.4157905044417405	Karonga district
-11.95341	33.367064	1.957231017237896	Mzimba disctrict
-11.632151	34.235546	1.3677132387498063	Nkhata Bay





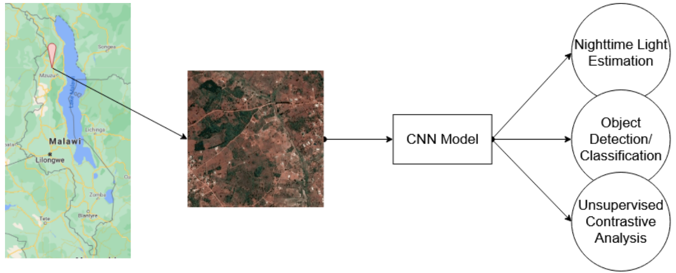
Indeed, we can't have a sufficient number of pairs of satellite images and ground truth poverty values. To obtain ground truth poverty values, surveyors has to question local population on their quality of life. This process is time consuming and expensive, that's probably why most of the country in africa have less than a thousand ground truth poverty value.
Plus, the latitude and longitude reported in the surveys are randomly shifted, up to five kilometers radius from the real place where the ground truth poverty value is measured. This shift is done to protect the anonimity of respondents. As a consequence, like illustrated on the right of the figure, the image we want to pair with the poverty ground truth may not represent the actual true location.

These challenges encourage researchers to develop advanced method for poverty prediction. We review three of them in the following.

Some Existing Methods

Transfer learning:

1) Define a problem related to poverty estimation

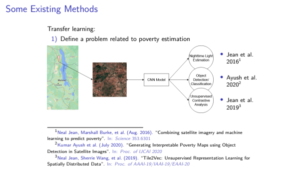


- Jean et al. 2016[1]
- Ayush et al. 2020[2]
- Jean et al. 2019[3]

[1] Neal Jean, Marshall Burke, et al. (Aug. 2016). "Combining satellite imagery and machine learning to predict poverty". In: *Science* 353.6301

[2] Kumar Ayush et al. (July 2020). "Generating Interpretable Poverty Maps using Object Detection in Satellite Images". In: *Proc. of IJCAI 2020*

[3] Neal Jean, Sherrie Wang, et al. (2019). "Tile2Vec: Unsupervised Representation Learning for Spatially Distributed Data". In: *Proc. of AAAI-19/IAAI-19/EAAI-20*

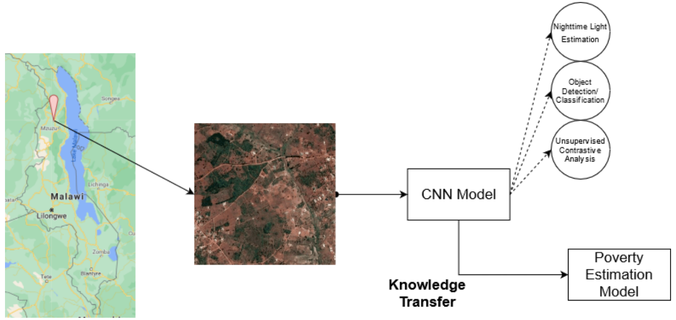


We focus on research works that use Transfer Learning. While it's difficult to have a large dataset of poverty values, obtaining large dataset of indicators closely related to poverty is possible. [HERE] The idea is first, to train a CNN on a problem related to poverty estimation. Several problem related to poverty were studied like Nighttime light estimation, Object detection and classification and unsupervised contrastive analysis.
[SLIDES]
Then, this knowledge is transfer with deep feature extraction, to a poverty estimation model, which consist in a regressor that outputs a scalar poverty estimation accoring to the input features given by the CNN.

Some Existing Methods

Transfer learning:

2) Transfer the knowlege

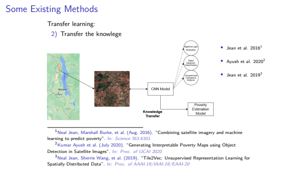


- Jean et al. 2016[1]
- Ayush et al. 2020[2]
- Jean et al. 2019[3]

[1] Neal Jean, Marshall Burke, et al. (Aug. 2016). "Combining satellite imagery and machine learning to predict poverty". In: *Science* 353.6301

[2] Kumar Ayush et al. (July 2020). "Generating Interpretable Poverty Maps using Object Detection in Satellite Images". In: *Proc. of IJCAI 2020*

[3] Neal Jean, Sherrie Wang, et al. (2019). "Tile2Vec: Unsupervised Representation Learning for Spatially Distributed Data". In: *Proc. of AAAI-19/IAAI-19/EAAI-20*



We focus on research works that use Transfer Learning. While it's difficult to have a large dataset of poverty values, obtaining large dataset of indicators closely related to poverty is possible. [HERE] The idea is first, to train a CNN on a problem related to poverty estimation. Several problem related to poverty were studied like Nighttime light estimation, Object detection and classification and unsupervised contrastive analysis.

[SLIDES]

Then, this knowledge is transfer with deep feature extraction, to a poverty estimation model, which consist in a regressor that outputs a scalar poverty estimation accoring to the input features given by the CNN.

Contributions: Assessing and Improve

Which of the three CNN models is the best for feature extraction?

Contributions: Assessing and Improve

Which of the three CNN models is the best for feature extraction?

Then, a natural question is: Among the three CNN model, which one is the best for extracting relevant features for poverty estimation ?
[SLIDES] As these work were made with several types of images, poverty values, CNN architecture and hyperparameters, it's difficult to assess and compare them. According to this, our work includes two contributions.
[SLIDES] As a first contribution, we aim to build a common framework for poverty prediction, that will allow us to fairly assess different existing poverty prediction models. It includes the same image type, the same poverty indicators, the same CNN architecture and the same hyperparameters.
[SLIDES] As this assessment allows us to better understand the weaknesses of such poverty prediction models, we propose some improvement on existing models that slightly ameliorate the prediction quality.

Contributions: Assessing and Improve

Which of the three CNN models is the best for feature extraction?

→ Different data providers and experimental settings prevent from fair comparison

Contributions: Assessing and Improve

Which of the three CNN models is the best for feature extraction?

→ Different data providers and experimental settings prevent from fair comparison

Then, a natural question is: Among the three CNN model, which one is the best for extracting relevant features for poverty estimation ?

[SLIDES] As these work were made with several types of images, poverty values, CNN architecture and hyperparameters, it's difficult to assess and compare them. According to this, our work includes two contributions.

[SLIDES] As a first contribution, we aim to build a common framework for poverty prediction, that will allow us to fairly assess different existing poverty prediction models. It includes the same image type, the same poverty indicators, the same CNN architecture and the same hyperparameters.

[SLIDES] As this assessment allows us to better understand the weaknesses of such poverty prediction models, we propose some improvement on existing models that slightly ameliorate the prediction quality.

Contributions: Assessing and Improve

Which of the three CNN models is the best for feature extraction?

→ Different data providers and experimental settings prevent from fair comparison

- Assessing the 3 methods
  ⟶ Requires a common framework:
  - Same image type
  - Same poverty indicator
  - Same CNN architecture
  - Same hyperparameters

Contributions: Assessing and Improve
Which of the three CNN models is the best for feature extraction?
→ Different data providers and experimental settings prevent from fair comparison
Assessing the 3 methods
⟶ Requires a common framework:
Same image type
Same poverty indicator
Same CNN architecture
Same hyperparameters

Then, a natural question is: Among the three CNN model, which one is the best for extracting relevant features for poverty estimation ?
[SLIDES] As these work were made with several types of images, poverty values, CNN architecture and hyperparameters, it's difficult to assess and compare them. According to this, our work includes two contributions.
[SLIDES] As a first contribution, we aim to build a common framework for poverty prediction, that will allow us to fairly assess different existing poverty prediction models. It includes the same image type, the same poverty indicators, the same CNN architecture and the same hyperparameters.
[SLIDES] As this assessment allows us to better understand the weaknesses of such poverty prediction models, we propose some improvement on existing models that slightly ameliorate the prediction quality.

Contributions: Assessing and Improve

Which of the three CNN models is the best for feature extraction?

→ Different data providers and experimental settings prevent from fair comparison

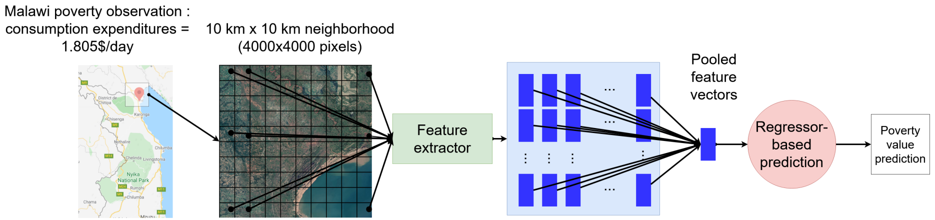
- Assessing the 3 methods
  ⟶ Requires a common framework:
  - Same image type
  - Same poverty indicator
  - Same CNN architecture
  - Same hyperparameters
- Extend existing models to improve the performances

Contributions: Assessing and Improve
Which of the three CNN models is the best for feature extraction?
→ Different data providers and experimental settings prevent from fair comparison
• Assessing the 3 methods
⟶ Requires a common framework:
• Same image type
• Same poverty indicator
• Same CNN architecture
• Same hyperparameters
• Extend existing models to improve the performances

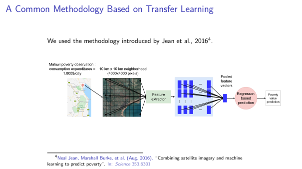
Then, a natural question is: Among the three CNN model, which one is the best for extracting relevant features for poverty estimation ?
[SLIDES] As these work were made with several types of images, poverty values, CNN architecture and hyperparameters, it's difficult to assess and compare them. According to this, our work includes two contributions.
[SLIDES] As a first contribution, we aim to build a common framework for poverty prediction, that will allow us to fairly assess different existing poverty prediction models. It includes the same image type, the same poverty indicators, the same CNN architecture and the same hyperparameters.
[SLIDES] As this assessment allows us to better understand the weaknesses of such poverty prediction models, we propose some improvement on existing models that slightly ameliorate the prediction quality.

A Common Methodology Based on Transfer Learning

We used the methodology introduced by Jean et al., 2016[4].



[4] Neal Jean, Marshall Burke, et al. (Aug. 2016). "Combining satellite imagery and machine learning to predict poverty". In: *Science* 353.6301

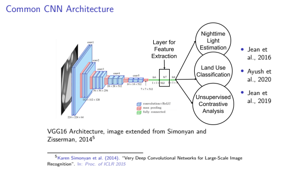


Now, let me introduce the framework methodology. As there is not enough data to build a deep CNN, the methods found in the litterature are based on transfer learning. This framework includes 2 models that needs to be trained, the feature extractor in green and the regressor in red. But for the moment, let us consider that they are trained and ready to use.

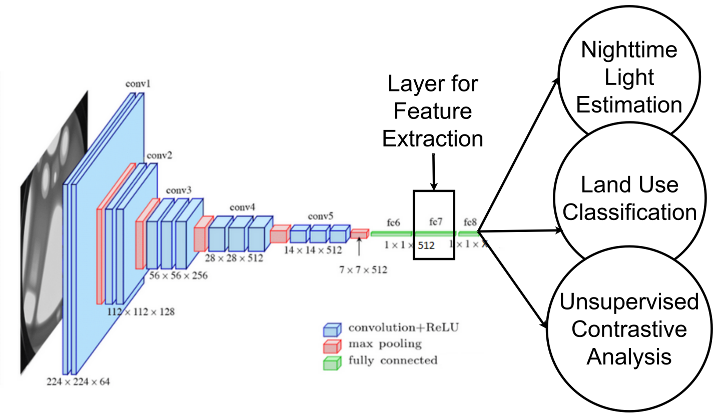
After selecting a place, we consider a 10km times 10km image centered in that place, doing so we are sure that the true location is in the image. As this image is large, it's divided into sub-images. All the sub images are then processed by a feature extractor, leading to a feature vector representation of the image. These feature vectors are the averaged and the resulting feature vector is processed by a Ridge Regression model, that give the poverty estimation.

The Ridge regression model is trained with pairs of feature vectors, given by the feature extractor and ground truth poverty values.

The transfer learning is perform with the feature exctractor. As it is trained on tasks closely related to poverty estimation, it's able to exctract relevant features of images. Let's focus on the featur extracor.



Common CNN Architecture



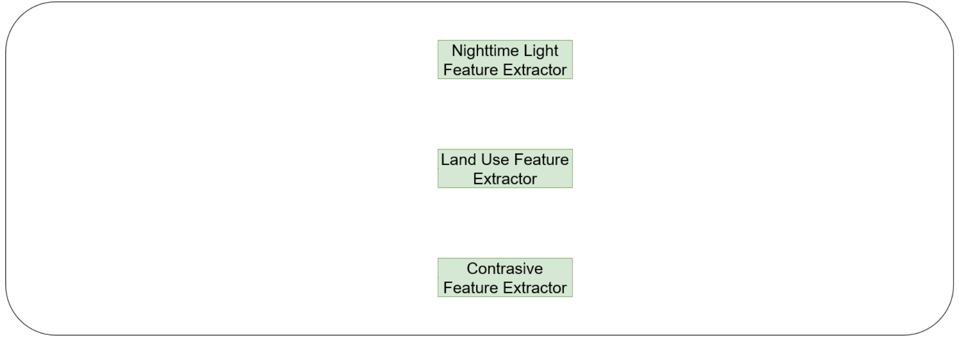
- Jean et al., 2016
- Ayush et al., 2020
- Jean et al., 2019

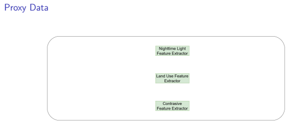
VGG16 Architecture, image extended from Simonyan and Zisserman, 2014[5]

[5] Karen Simonyan et al. (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proc. of ICLR 2015*

As we wan to assess the 3 methods previously mentionned, we built 3 feature extractors, with the same architecture. This is a classic VGG16 architecture as it commonly used for its robustness over different types of tasks. As you can observe, The feature vectors are generated with a specific deep hidden layer of the CNN. The first feature extractor is based on Jean et al, 2016 which is trained to Nighttime light values. The second feature extractor is a CNN that predicts Land use classes, assuming that the land use is related to poverty. Note here that we simplify the detection and classification task made in Ayush et al. with a single classification task. The third feature extractor is trained to build similar feature vector representation for images that look similar, and dissimilar feature vector representation for images that look dissimilar. Its based on Jean et al.'s work in 2019.
[SLIDES]
Now, let me introduce the data used to train the feature extractors.

Proxy Data





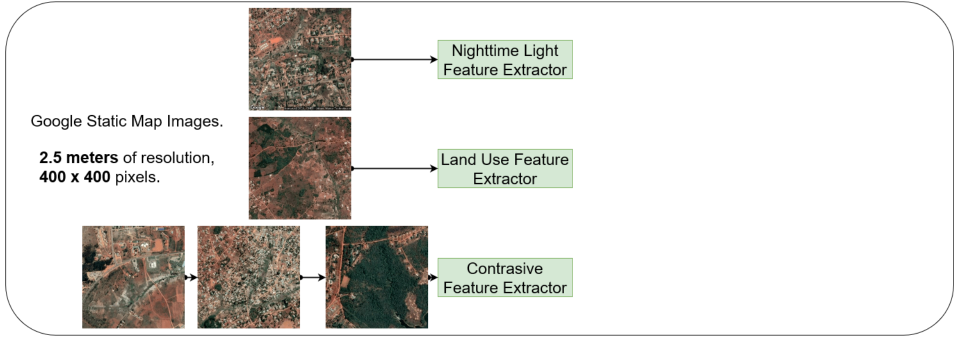
As input, it's important to have the same image type, doing so, image type has no influence on the relative results of the feature extractors. [SLIDES] The images used to train the feature extractors are provided by the Google Static Map API, 2.5 meters of resolution and 400 times 400 pixels. It covers a 1 squared kilometer area, roughly the size of a village. the feature extractors differs from the labels they are trained on.

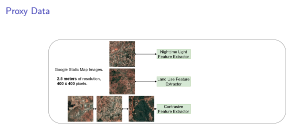
[SLIDES] For the Night-time light model, Night-time light labels are provided by the Earth Observation Group, and corresponds to Low, medium and High Night-time light value.

[SLIDES] For the Land-use model, Land use labels are provided by the EuroSat Dataset. It consists of Georeferenced satellite images labeld into 10 land use classes.

[SLIDES] Finally, for the contrastive model, there is no need of labels, but we need to sample triplet of images. Each triplet is composed of an anchor, neighbor and distant image. The anchor image is randomly sample, the neighbor image is sample in a close neighborhood of the anchor image, and the distant image is sample in a larger neighborhood of the anchor image. Then, the CNN will learn to make similar representation for the anchor and neighbor image, and dissimilar representation for the distant image, thanks to a custom loss function.

Proxy Data





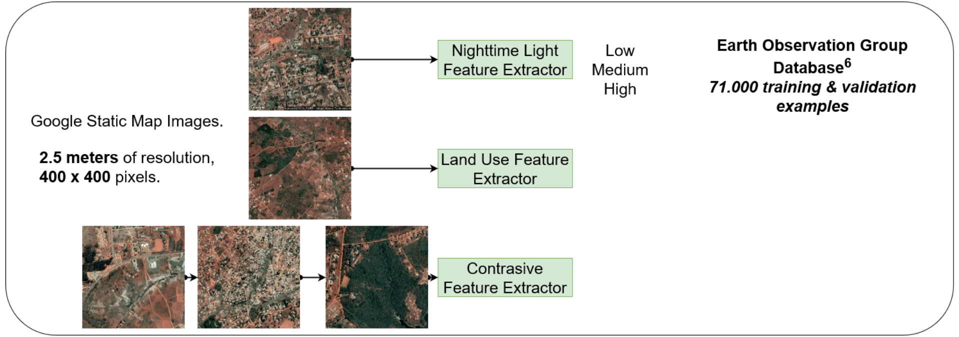
As input, it's important to have the same image type, doing so, image type has no influence on the relative results of the feature extractors. [SLIDES] The images used to train the feature extractors are provided by the Google Static Map API, 2.5 meters of resolution and 400 times 400 pixels. It covers a 1 squared kilometer area, roughly the size of a village. the feature extractors differs from the labels they are trained on.

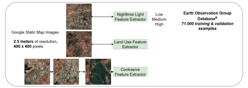
[SLIDES] For the Night-time light model, Night-time light labels are provided by the Earth Observation Group, and corresponds to Low, medium and High Night-time light value.

[SLIDES] For the Land-use model, Land use labels are provided by the EuroSat Dataset. It consists of Georeferenced satellite images labeld into 10 land use classes.

[SLIDES] Finally, for the contrastive model, there is no need of labels, but we need to sample triplet of images. Each triplet is composed of an anchor, neighbor and distant image. The anchor image is randomly sample, the neighbor image is sample in a close neighborhood of the anchor image, and the distant image is sample in a larger neighborhood of the anchor image. Then, the CNN will learn to make similar representation for the anchor and neighbor image, and dissimilar representation for the distant image, thanks to a custom loss function.

Proxy Data





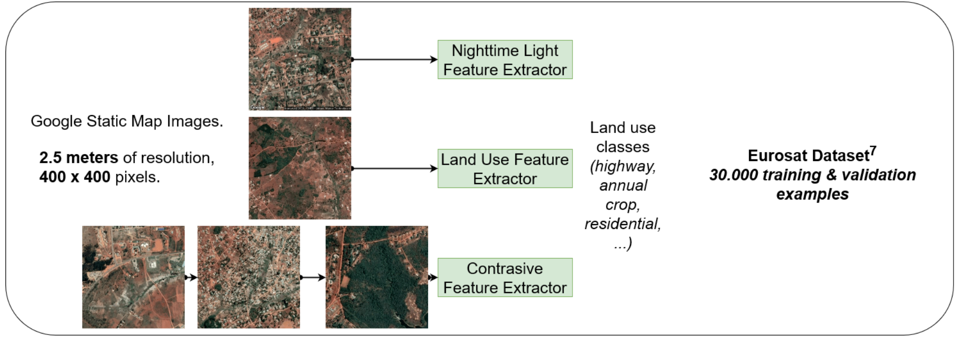
As input, it's important to have the same image type, doing so, image type has no influence on the relative results of the feature extractors. [SLIDES] The images used to train the feature extractors are provided by the Google Static Map API, 2.5 meters of resolution and 400 times 400 pixels. It covers a 1 squared kilometer area, roughly the size of a village. the feature extractors differs from the labels they are trained on.

[SLIDES] For the Night-time light model, Night-time light labels are provided by the Earth Observation Group, and corresponds to Low, medium and High Night-time light value.

[SLIDES] For the Land-use model, Land use labels are provided by the EuroSat Dataset. It consists of Georeferenced satellite images labeld into 10 land use classes.

[SLIDES] Finally, for the contrastive model, there is no need of labels, but we need to sample triplet of images. Each triplet is composed of an anchor, neighbor and distant image. The anchor image is randomly sample, the neighbor image is sample in a close neighborhood of the anchor image, and the distant image is sample in a larger neighborhood of the anchor image. Then, the CNN will learn to make similar representation for the anchor and neighbor image, and dissimilar representation for the distant image, thanks to a custom loss function.

Proxy Data



Proxy Data

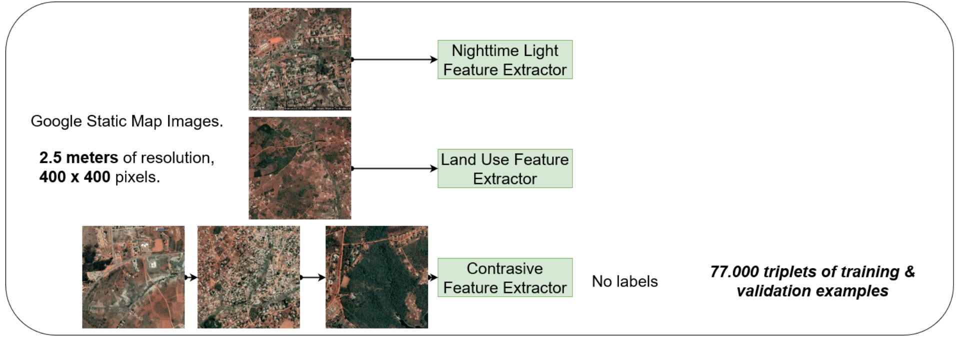
As input, it's important to have the same image type, doing so, image type has no influence on the relative results of the feature extractors. [SLIDES] The images used to train the feature extractors are provided by the Google Static Map API, 2.5 meters of resolution and 400 times 400 pixels. It covers a 1 squared kilometer area, roughly the size of a village. the feature extractors differs from the labels they are trained on.

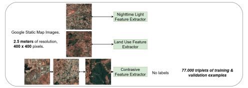
[SLIDES] For the Night-time light model, Night-time light labels are provided by the Earth Observation Group, and corresponds to Low, medium and High Night-time light value.

[SLIDES] For the Land-use model, Land use labels are provided by the EuroSat Dataset. It consists of Georeferenced satellite images labeld into 10 land use classes.

[SLIDES] Finally, for the contrastive model, there is no need of labels, but we need to sample triplet of images. Each triplet is composed of an anchor, neighbor and distant image. The anchor image is randomly sample, the neighbor image is sample in a close neighborhood of the anchor image, and the distant image is sample in a larger neighborhood of the anchor image. Then, the CNN will learn to make similar representation for the anchor and neighbor image, and dissimilar representation for the distant image, thanks to a custom loss function.

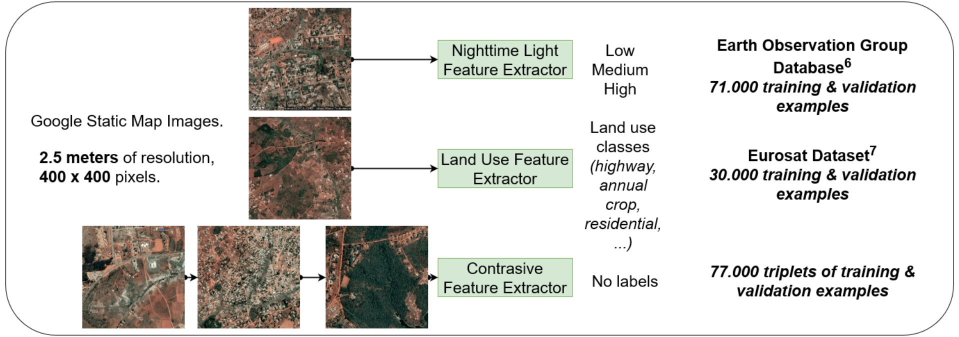
Proxy Data





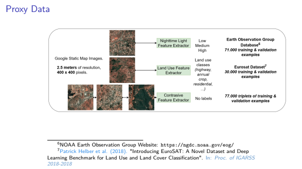
As input, it's important to have the same image type, doing so, image type has no influence on the relative results of the feature extractors. [SLIDES] The images used to train the feature extractors are provided by the Google Static Map API, 2.5 meters of resolution and 400 times 400 pixels. It covers a 1 squared kilometer area, roughly the size of a village. the feature extractors differs from the labels they are trained on.
[SLIDES] For the Night-time light model, Night-time light labels are provided by the Earth Observation Group, and corresponds to Low, medium and High Night-time light value.
[SLIDES] For the Land-use model, Land use labels are provided by the EuroSat Dataset. It consists of Georeferenced satellite images labeld into 10 land use classes.
[SLIDES] Finally, for the contrastive model, there is no need of labels, but we need to sample triplet of images. Each triplet is composed of an anchor, neighbor and distant image. The anchor image is randomly sample, the neighbor image is sample in a close neighborhood of the anchor image, and the distant image is sample in a larger neighborhood of the anchor image. Then, the CNN will learn to make similar representation for the anchor and neighbor image, and dissimilar representation for the distant image, thanks to a custom loss function.

Proxy Data



[6] NOAA Earth Observation Group Website: `https://ngdc.noaa.gov/eog/`

[7] Patrick Helber et al. (2018). "Introducing EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification". *In: Proc. of IGARSS 2018-2018*



As input, it's important to have the same image type, doing so, image type has no influence on the relative results of the feature extractors. [SLIDES] The images used to train the feature extractors are provided by the Google Static Map API, 2.5 meters of resolution and 400 times 400 pixels. It covers a 1 squared kilometer area, roughly the size of a village. the feature extractors differs from the labels they are trained on.

[SLIDES] For the Night-time light model, Night-time light labels are provided by the Earth Observation Group, and corresponds to Low, medium and High Night-time light value.

[SLIDES] For the Land-use model, Land use labels are provided by the EuroSat Dataset. It consists of Georeferenced satellite images labeld into 10 land use classes.

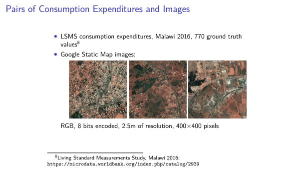
[SLIDES] Finally, for the contrastive model, there is no need of labels, but we need to sample triplet of images. Each triplet is composed of an anchor, neighbor and distant image. The anchor image is randomly sample, the neighbor image is sample in a close neighborhood of the anchor image, and the distant image is sample in a larger neighborhood of the anchor image. Then, the CNN will learn to make similar representation for the anchor and neighbor image, and dissimilar representation for the distant image, thanks to a custom loss function.

Pairs of Consumption Expenditures and Images

- LSMS consumption expenditures, Malawi 2016, 770 ground truth values[8]
- Google Static Map images:

RGB, 8 bits encoded, 2.5m of resolution, 400×400 pixels

[8] Living Standard Measurements Study, Malawi 2016: https://microdata.worldbank.org/index.php/catalog/2939



We use as poverty indicator the consumption expenditures given in LSMS surveys of Malawi in 2016. This data is collected at the household level. We used the 770 clusters referenced in the dataset to train and test the models. We used images provided by the Google Static Map API, as it was done for the proxy image data. Here are some examples, depicting the city of Mzuzu in Malawi.

Now, let me introduce the experimental settings for assessing the methods

Experimental Settings

1. Feature extractor training + validation

Experimental Settings

1. Feature extractor training + validation

We perfor a traning and validation of the feature extractor. Once the results are consistent,
[SLIDES]
We perform feature extraction for the seven hundred and seventy images coresponding to ground truth poverty values.
[SLIDES]
To have consistent results, we perform a 10 fold cross validation over the 770 deature vectors and poverty values.
[SLIDES]
We compute the average test R2 score, where R2 score is one minus the sum of squared difference of predicted and observed poverty values, over the sum of squared difference of the observed poverty values and the average poverty value. We want the difference between the true and predicted values to be small, so we want R2 to be close to 1.

Experimental Settings

1. Feature extractor training + validation
2. Feature extraction for all 770 images

Experimental Settings
1. Feature extractor training + validation
2. Feature extraction for all 770 images

We perfor a traning and validation of the feature extractor. Once the results are consistent,
[SLIDES]
We perform feature extraction for the seven hundred and seventy images coresponding to ground truth poverty values.
[SLIDES]
To have consistent results, we perform a 10 fold cross validation over the 770 deature vectors and poverty values.
[SLIDES]
We compute the average test R2 score, where R2 score is one minus the sum of squared difference of predicted and observed poverty values, over the sum of squared difference of the observed poverty values and the average poverty value. We want the difference between the true and predicted values to be small, so we want R2 to be close to 1.

Experimental Settings

1. Feature extractor training + validation
2. Feature extraction for all 770 images
3. Regression with 10-fold cross validation over the 770 extracted feature vectors

Experimental Settings

1. Feature extractor training + validation
2. Feature extraction for all 770 images
3. Regression with 10-fold cross validation over the 770 extracted feature vectors

We perfor a traning and validation of the feature extractor. Once the results are consistent,
[SLIDES]
We perform feature extraction for the seven hundred and seventy images coresponding to ground truth poverty values.
[SLIDES]
To have consistent results, we perform a 10 fold cross validation over the 770 deature vectors and poverty values.
[SLIDES]
We compute the average test R2 score, where R2 score is one minus the sum of squared difference of predicted and observed poverty values, over the sum of squared difference of the observed poverty values and the average poverty value. We want the difference between the true and predicted values to be small, so we want R2 to be close to 1.

Experimental Settings

1. Feature extractor training + validation
2. Feature extraction for all 770 images
3. Regression with 10-fold cross validation over the 770 extracted feature vectors
4. Assessing with R^2 score:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

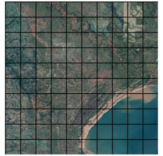
Experimental Settings

1. Feature extractor training + validation
2. Feature extraction for all 770 images
3. Regression with 10-fold cross validation over the 770 extracted feature vectors
4. Assessing with R^2 score:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

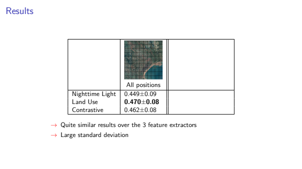
We perfor a traning and validation of the feature extractor. Once the results are consistent,
[SLIDES]
We perform feature extraction for the seven hundred and seventy images coresponding to ground truth poverty values.
[SLIDES]
To have consistent results, we perform a 10 fold cross validation over the 770 deature vectors and poverty values.
[SLIDES]
We compute the average test R2 score, where R2 score is one minus the sum of squared difference of predicted and observed poverty values, over the sum of squared difference of the observed poverty values and the average poverty value. We want the difference between the true and predicted values to be small, so we want R2 to be close to 1.

Results

	 All positions	
Nighttime Light	0.449 ± 0.09	
Land Use	**0.470 ± 0.08**	
Contrastive	0.462 ± 0.08	

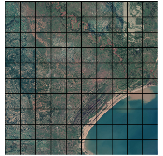
→ Quite similar results over the 3 feature extractors

→ Large standard deviation



For a given feature extraction, the results are similar between the three feature extractors, the maximum differences is about 0.02.
Also, note the high standard deviation over the validation folds, which prevent us from ranking the models.
Here, the results are consistent with what is obtained in the articles we are assessing.
[SLIDES]
Then, We compare an other feed-forward approaches where the feature vector is only computed for the image at the anonymized position
We can observe that considering only the anonymized position is reducing the prediction quality. Including all the position captures the image were the true location is, that can explain the better results. But, comparing to the single anonymized position, it also includes geographical context, that may help to build a better feature representation.

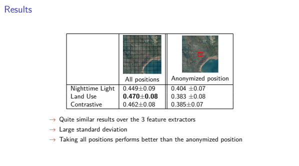
Results

	 All positions	 Anonymized position
Nighttime Light	0.449±0.09	0.404 ±0.07
Land Use	**0.470±0.08**	0.383 ±0.08
Contrastive	0.462±0.08	0.385±0.07

→ Quite similar results over the 3 feature extractors

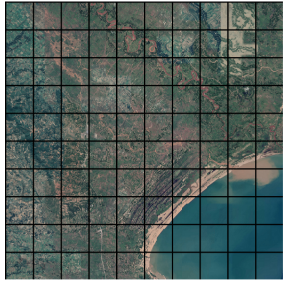
→ Large standard deviation

→ Taking all positions performs better than the anonymized position



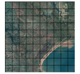
For a given feature extraction, the results are similar between the three feature extractors, the maximum differences is about 0.02.
Also, note the high standard deviation over the validation folds, which prevent us from ranking the models.
Here, the results are consistent with what is obtained in the articles we are assessing.
[SLIDES]
Then, We compare an other feed-forward approaches where the feature vector is only computed for the image at the anonymized position
We can observe that considering only the anonymized position is reducing the prediction quality. Including all the position captures the image were the true location is, that can explain the better results. But, comparing to the single anonymized position, it also includes geographical context, that may help to build a better feature representation.

Improvement (1): Reducing Spatial Noise Effects



WorldPop: https://www.worldpop.org/geodata/summary?id=123

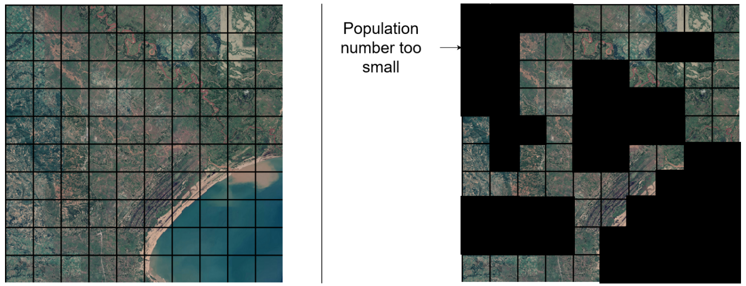
Improvement (1): Reducing Spatial Noise Effects



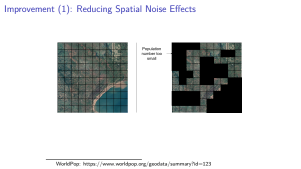
WorldPop: https://www.worldpop.org/geodata/summary?id=123

As the geographic discplacement should be a real insight for the prediction quality, we propose to include grid-cell selection in the neighborhood of the anonymized position.
[SLIDES]
As shown on the figure, for each neighborhood, Instead of selecting the complete neigborhood, we select the cells where the true position is more likely to be. For each grid cell, we compute it's Night-time light value and it's average population number. The cells are then ranked by decreasing order, and we select the first 20 cells that have the most important Night-time light value and population number.

Improvement (1): Reducing Spatial Noise Effects

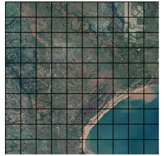


WorldPop: https://www.worldpop.org/geodata/summary?id=123

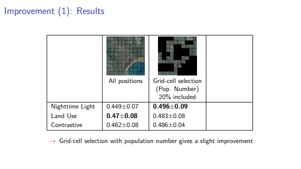


As the geographic discplacement should be a real insight for the prediction quality, we propose to include grid-cell selection in the neighborhood of the anonymized position.
[SLIDES]
As shown on the figure, for each neighborhood, Instead of selecting the complete neigborhood, we select the cells where the true position is more likely to be. For each grid cell, we compute it's Night-time light value and it's average population number. The cells are then ranked by decreasing order, and we select the first 20 cells that have the most important Night-time light value and population number.

Improvement (1): Results

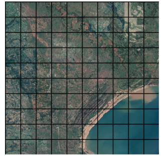
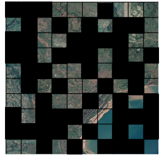
	 All positions	 Grid-cell selection (Pop. Number) 20% included	
Nighttime Light	0.449±0.07	**0.496±0.09**	
Land Use	**0.47±0.08**	0.483±0.08	
Contrastive	0.462±0.08	0.486±0.04	

→ Grid-cell selection with population number gives a slight improvement



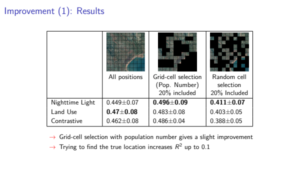
When we compare no selection (which means taking all position) and grid-cell selection, the results are slightly better for grid-cell selection but there is a high standard deviation. At least, we can say that we obtain similar performances, but with 80 percent less images.
[SLIDES]
Also, when we perform a grid cell selection that returns the same amount of area, random selection, that probably not include the true location, is significantly lower. This strongly suggests that finding back the true location is essential for poverty estimation.

Improvement (1): Results

	 All positions	 Grid-cell selection (Pop. Number) 20% included	 Random cell selection 20% Included
Nighttime Light	0.449±0.07	**0.496±0.09**	**0.411±0.07**
Land Use	**0.47±0.08**	0.483±0.08	0.403±0.05
Contrastive	0.462±0.08	0.486±0.04	0.388±0.05

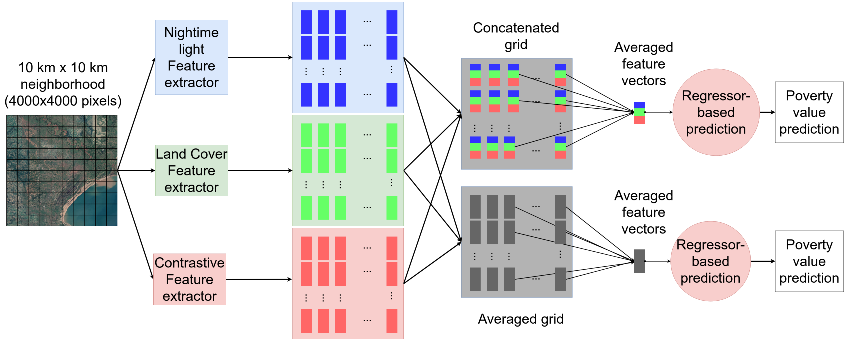
→ Grid-cell selection with population number gives a slight improvement

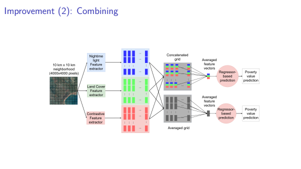
→ Trying to find the true location increases R^2 up to 0.1



When we compare no selection (which means taking all position) and grid-cell selection, the results are slightly better for grid-cell selection but there is a high standard deviation. At least, we can say that we obtain similar performances, but with 80 percent less images.
[SLIDES]
Also, when we perform a grid cell selection that returns the same amount of area, random selection, that probably not include the true location, is significantly lower. This strongly suggests that finding back the true location is essential for poverty estimation.

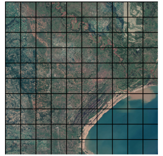
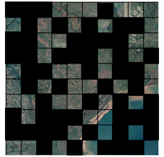
Improvement (2): Combining





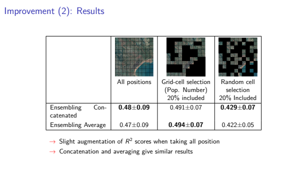
Then, it's natural to try the three models together, to see if it's improving performances. To combine the 3 models, we proceed as illustrated. Each Neighborhood is processed by the 3 feature extractors, and generate a grid of feature vector. these 3 grids are either concatenated or averaged. The concatenated grid or averaged grid are average, leading to a feature vector. Finally this feature vector is the inpour of a Ridge regresion model.

Improvement (2): Results

	 All positions	 Grid-cell selection (Pop. Number) 20% included	 Random cell selection 20% Included
Ensembling Concatenated	**0.48±0.09**	0.491±0.07	**0.429±0.07**
Ensembling Average	0.47±0.09	**0.494±0.07**	0.422±0.05

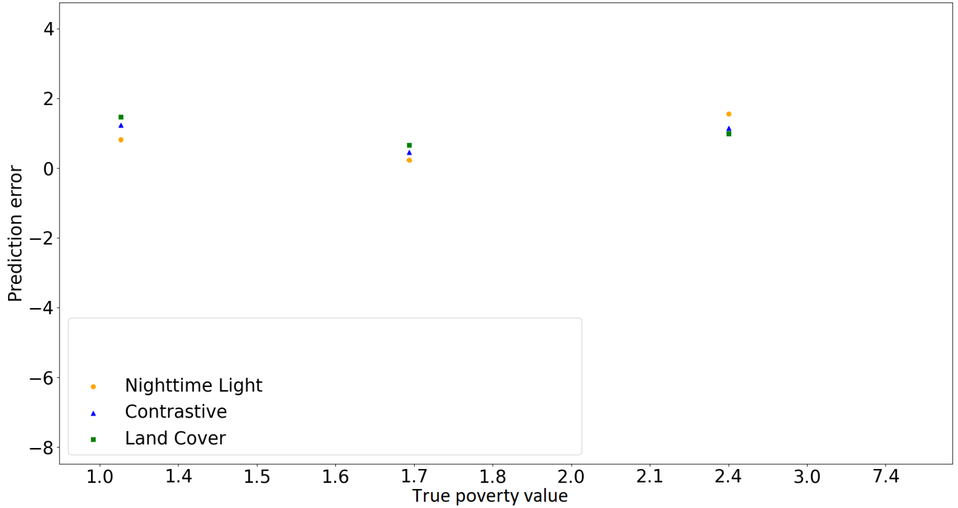
→ Slight augmentation of R^2 scores when taking all position

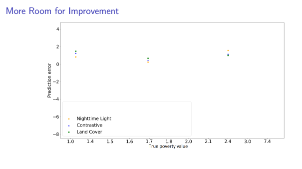
→ Concatenation and averaging give similar results



The performances augments slightly when taking all the positions compared to previous methods, but sitll, there is a high standard deviation. Else, the performances are equal to previous methods, [HERE] with almost no differences between the two combining approaches. But, these are quite simple combination methods. We experiment that advanced combination methods could give significantly better results.

More Room for Improvement





This figure plots the prediction error for a ground truth poverty value made by the 3 methods, over a validation fold. On the X axis, there are the ground truth values, supposed to be predicted by the models. Note that this axis is non linear. Y axis denotes the prediction error.

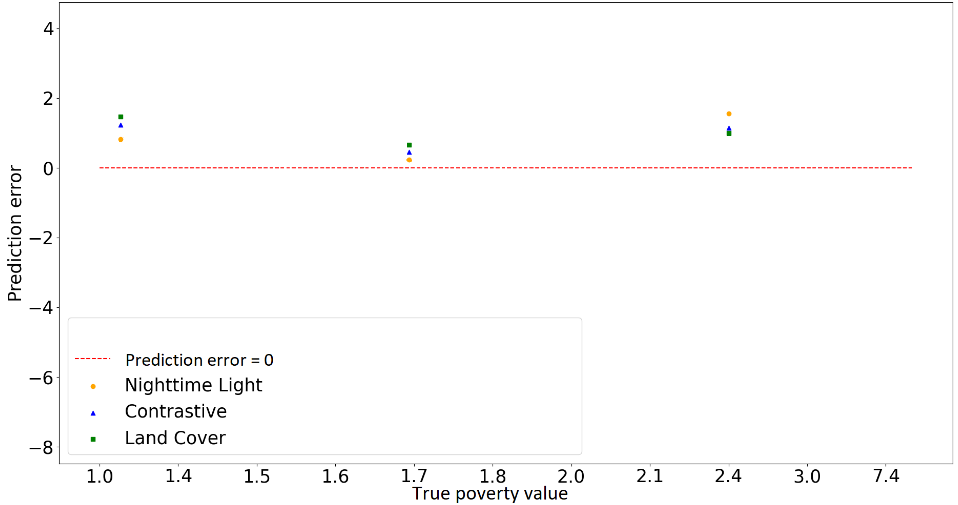
[SLIDES]

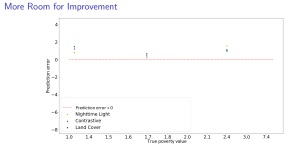
We can display the line where the prediction error is zero. Let's consider the three points on the left side of the figure. the true poverty value the models were supposed to predict is between 1 and 1.4. Among the three prediction, we can see that it's the nighttime light model, in yellow that makes the smallest error.

[SLIES] For each ground truth value, we have three prediction, one for each model. Among the three prediciton, we select prediciton that makes the smallest error. Then, we connext the selected points with a black line.

[SLIDES] After this process, we found that choosing the best prediction among the three model for each ground truth value leads to R2 equal 0.6. So, there exists a combination of the three models that can performs better than each model separately

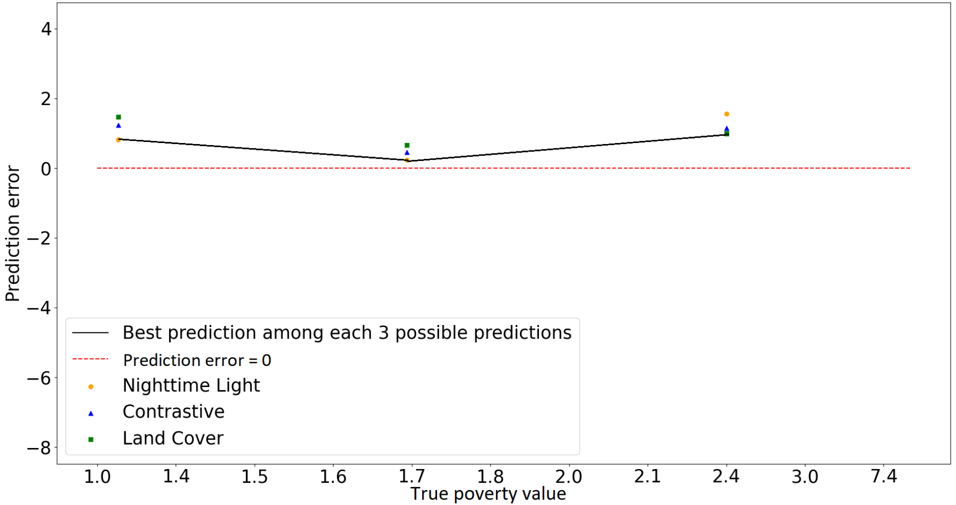
More Room for Improvement

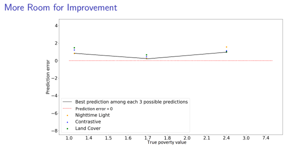




This figure plots the prediction error for a ground truth poverty value made by the 3 methods, over a validation fold. On the X axis, there are the ground truth values, supposed to be predicted by the models. Note that this axis is non linear. Y axis denotes the prediction error.
[SLIDES]
We can display the line where the prediction error is zero. Let's consider the three points on the left side of the figure. the true poverty value the models were supposed to predict is between 1 and 1.4. Among the three prediction, we can see that it's the nighttime light model, in yellow that makes the smallest error.
[SLIES] For each ground truth value, we have three prediction, one for each model. Among the three prediciton, we select prediciton that makes the smallest error. Then, we connext the selected points with a black line.
[SLIDES] After this process, we found that choosing the best prediction among the three model for each ground truth value leads to R2 equal 0.6. So, there exists a combination of the three models that can performs better than each model separately

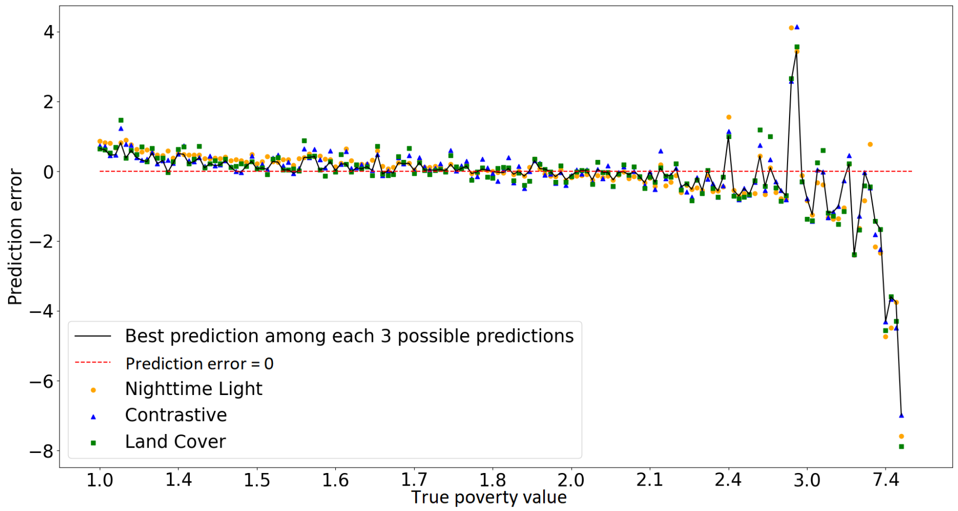
More Room for Improvement



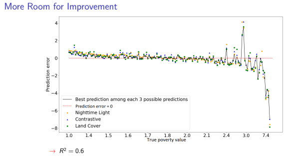


This figure plots the prediction error for a ground truth poverty value made by the 3 methods, over a validation fold. On the X axis, there are the ground truth values, supposed to be predicted by the models. Note that this axis is non linear. Y axis denotes the prediction error.
[SLIDES]
We can display the line where the prediction error is zero. Let's consider the three points on the left side of the figure. the true poverty value the models were supposed to predict is between 1 and 1.4. Among the three prediction, we can see that it's the nighttime light model, in yellow that makes the smallest error.
[SLIES] For each ground truth value, we have three prediction, one for each model. Among the three prediciton, we select prediciton that makes the smallest error. Then, we connext the selected points with a black line.
[SLIDES] After this process, we found that choosing the best prediction among the three model for each ground truth value leads to R2 equal 0.6. So, there exists a combination of the three models that can performs better than each model separately

More Room for Improvement



$\rightarrow R^2 = 0.6$



This figure plots the prediction error for a ground truth poverty value made by the 3 methods, over a validation fold. On the X axis, there are the ground truth values, supposed to be predicted by the models. Note that this axis is non linear. Y axis denotes the prediction error.

[SLIDES]

We can display the line where the prediction error is zero. Let's consider the three points on the left side of the figure. the true poverty value the models were supposed to predict is between 1 and 1.4. Among the three prediction, we can see that it's the nighttime light model, in yellow that makes the smallest error.

[SLIES] For each ground truth value, we have three prediction, one for each model. Among the three prediciton, we select prediciton that makes the smallest error. Then, we connext the selected points with a black line.

[SLIDES] After this process, we found that choosing the best prediction among the three model for each ground truth value leads to R2 equal 0.6. So, there exists a combination of the three models that can performs better than each model separately

Summary of Key Results

- High standard deviation prevents from ranking the models
- The spatial noise impacts the learning process
- Combining the models slightly improves the prediction quality
- Advanced combining methods seem promising

Summary of Key Results

- High standard deviation prevents from ranking the models
- The spatial noise impacts the learning process
- Combining the models slightly improves the prediction quality
- Advanced combining methods seem promising

An important results illustrated here is that the spatial noise impacts the learning process, but considering the entire neighborhood where the true location should be and applying grid cell selction leads similar performances, Due to te second point. Indeed the dataset of poverty values is very small, and the distribution of train and test sets are very diffirent, leading to a high standard deviation. Combining the models slightly improves the prediction quality, but here again, it's difficult to conclude because of the high standard deviation. Finally, as we try simple combination methods, and our last results sugessts that more advanced combinig method should improve the performance.

Open Questions

- Studying the entire Africa increases the number of ground truth values

[6] Christopher Yeh et al. (May 2020). "Using publicly available satellite imagery and deep learning to understand economic well-being in Africa". In: *Nature Communications*

[7] Kamwoo Lee et al. (Sept. 2020). "High-Resolution Poverty Maps in Sub-Saharan Africa". In: *Unpublished*

[8] Lukas Kondmann et al. (2020). "Measuring Changes in Poverty with Deep Learning and Satellite Images". In: *Proc. of ICLR 2020*

Open Questions

- Studying the entire Africa increases the number of ground truth values

[6] Christopher Yeh et al. (May 2020). "Using publicly available satellite imagery and deep learning to understand economic well-being in Africa". In: Nature Communications

[7] Kamwoo Lee et al. (Sept. 2020). "High-Resolution Poverty Maps in Sub-Saharan Africa". In: Unpublished

[8] Lukas Kondmann et al. (2020). "Measuring Changes in Poverty with Deep Learning and Satellite Images". In: Proc. of ICLR 2020

Recent work experiment that enlarging the study area to the entier africa,
[SLIDES] thus considering more ground truth poverty values allow to perform a direct training, as done in Yeh et al.
[SLIDES]
Also, a new methodology that uses transfer learning reaches very high R2 score, upt to 0 point ninty five.
[SLIDES] Poverty prediction became in a couple of years a very popular research topic. But, some works needs to be done concerning satellite image times series and poverty. For example is time, and in particular the evolution of changes noticed in a satellite image time serie, a good feature for poverty prediciton? A more important question is about forecasting a poverty value, are we able to predict poverty in the future? These seem to be challenging questions, as exposed in Kondmann et al.

Open Questions

- Studying the entire Africa increases the number of ground truth values
  - $\rightarrow$ A direct method is possible, with Landsat 7 images[6]

[6] Christopher Yeh et al. (May 2020). "Using publicly available satellite imagery and deep learning to understand economic well-being in Africa". In: *Nature Communications*

[7] Kamwoo Lee et al. (Sept. 2020). "High-Resolution Poverty Maps in Sub-Saharan Africa". In: *Unpublished*

[8] Lukas Kondmann et al. (2020). "Measuring Changes in Poverty with Deep Learning and Satellite Images". In: *Proc. of ICLR 2020*

Open Questions
- Studying the entire Africa increases the number of ground truth values
  - → A direct method is possible, with Landsat 7 images[6]

Recent work experiment that enlarging the study area to the entier africa,
[SLIDES] thus considering more ground truth poverty values allow to perform a direct training, as done in Yeh et al.
[SLIDES]
Also, a new methodology that uses transfer learning reaches very high R2 score, upt to 0 point ninty five.
[SLIDES] Poverty prediction became in a couple of years a very popular research topic. But, some works needs to be done concerning satellite image times series and poverty. For example is time, and in particular the evolution of changes noticed in a satellite image time serie, a good feature for poverty prediciton? A more important question is about forecasting a poverty value, are we able to predict poverty in the future? These seem to be challenging questions, as exposed in Kondmann et al.

Open Questions

- Studying the entire Africa increases the number of ground truth values
  - → A direct method is possible, with Landsat 7 images[6]
  - → Transfer learning methods are still a good option[7]

[6] Christopher Yeh et al. (May 2020). “Using publicly available satellite imagery and deep learning to understand economic well-being in Africa”. In: *Nature Communications*

[7] Kamwoo Lee et al. (Sept. 2020). “High-Resolution Poverty Maps in Sub-Saharan Africa”. In: *Unpublished*

[8] Lukas Kondmann et al. (2020). “Measuring Changes in Poverty with Deep Learning and Satellite Images”. In: *Proc. of ICLR 2020*

Recent work experiment that enlarging the study area to the entier africa,
[SLIDES] thus considering more ground truth poverty values allow to perform a direct training, as done in Yeh et al.
[SLIDES]
Also, a new methodology that uses transfer learning reaches very high R2 score, upt to 0 point ninty five.
[SLIDES] Poverty prediction became in a couple of years a very popular research topic. But, some works needs to be done concerning satellite image times series and poverty. For example is time, and in particular the evolution of changes noticed in a satellite image time serie, a good feature for poverty prediciton? A more important question is about forecasting a poverty value, are we able to predict poverty in the future? These seem to be challenging questions, as exposed in Kondmann et al.

Open Questions

- Studying the entire Africa increases the number of ground truth values
  - $\rightarrow$ A direct method is possible, with Landsat 7 images[6]
  - $\rightarrow$ Transfer learning methods are still a good option[7]

- Is time a good feature for poverty prediction? Is forecasting possible?
  - $\rightarrow$ The problem is set up in Kondmann et al.[8], and seems challenging

[6] Christopher Yeh et al. (May 2020). "Using publicly available satellite imagery and deep learning to understand economic well-being in Africa". In: *Nature Communications*

[7] Kamwoo Lee et al. (Sept. 2020). "High-Resolution Poverty Maps in Sub-Saharan Africa". In: *Unpublished*

[8] Lukas Kondmann et al. (2020). "Measuring Changes in Poverty with Deep Learning and Satellite Images". In: *Proc. of ICLR 2020*

Recent work experiment that enlarging the study area to the entier africa,
[SLIDES] thus considering more ground truth poverty values allow to perform a direct training, as done in Yeh et al.
[SLIDES]
Also, a new methodology that uses transfer learning reaches very high R2 score, upt to 0 point ninty five.
[SLIDES] Poverty prediction became in a couple of years a very popular research topic. But, some works needs to be done concerning satellite image times series and poverty. For example is time, and in particular the evolution of changes noticed in a satellite image time serie, a good feature for poverty prediciton? A more important question is about forecasting a poverty value, are we able to predict poverty in the future? These seem to be challenging questions, as exposed in Kondmann et al.

Thank you!

Any question?

Thank you!
Any question?

Feel free to ask your question, I'm ready to answer.