

510



AN INITIATIVE OF
THE NETHERLANDS
RED CROSS

AUTOMATED DAMAGE ASSESSMENT

Jacopo Margutti, Data Scientist, PhD

ML for Earth System Modelling and Analytics workshop 03-04 May 2021

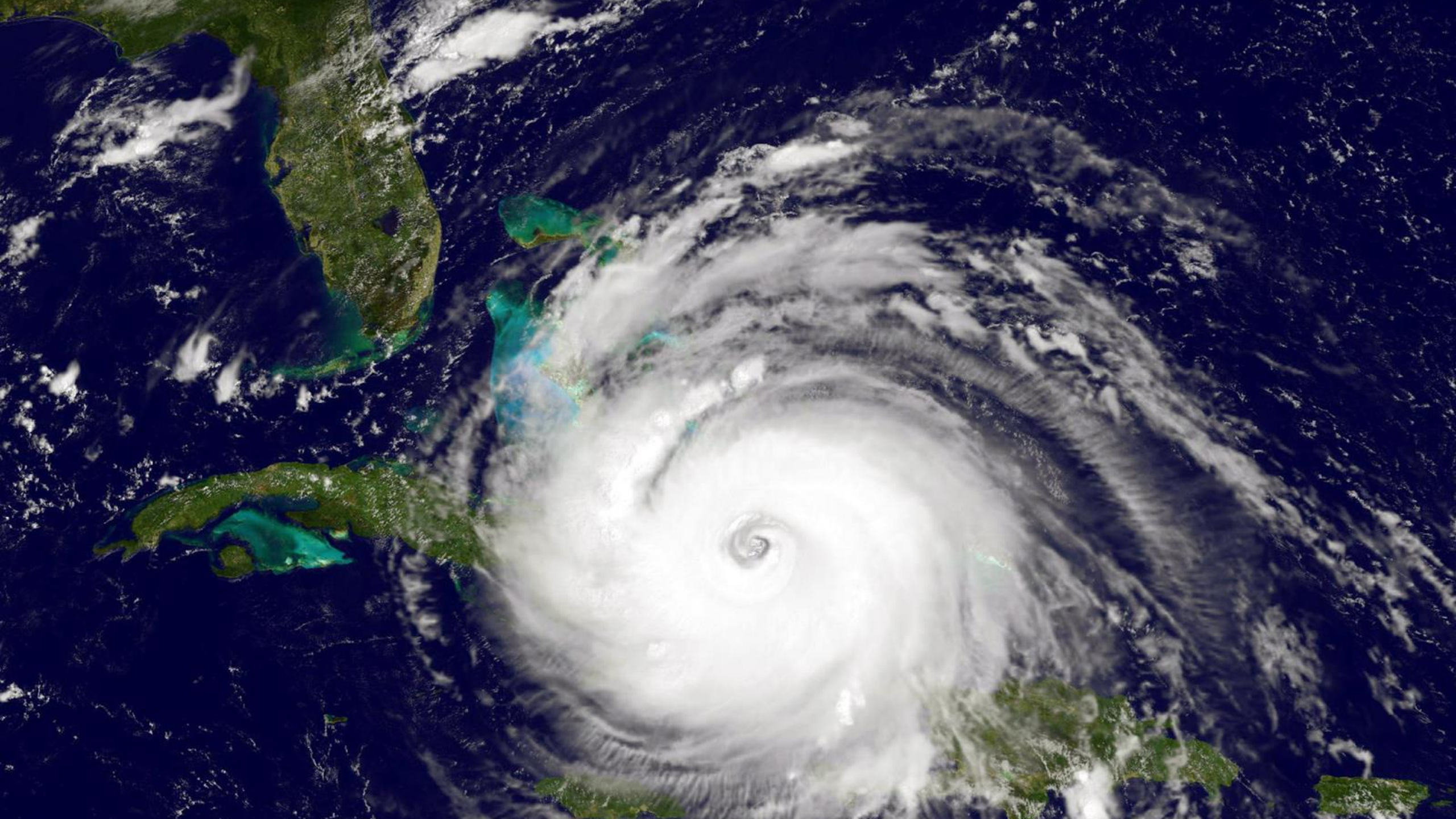
OUTLINE

WHY AUTOMATED DAMAGE ASSESSMENT

HOW WE DO IT




WHY AUTOMATED DAMAGE ASSESSMENT

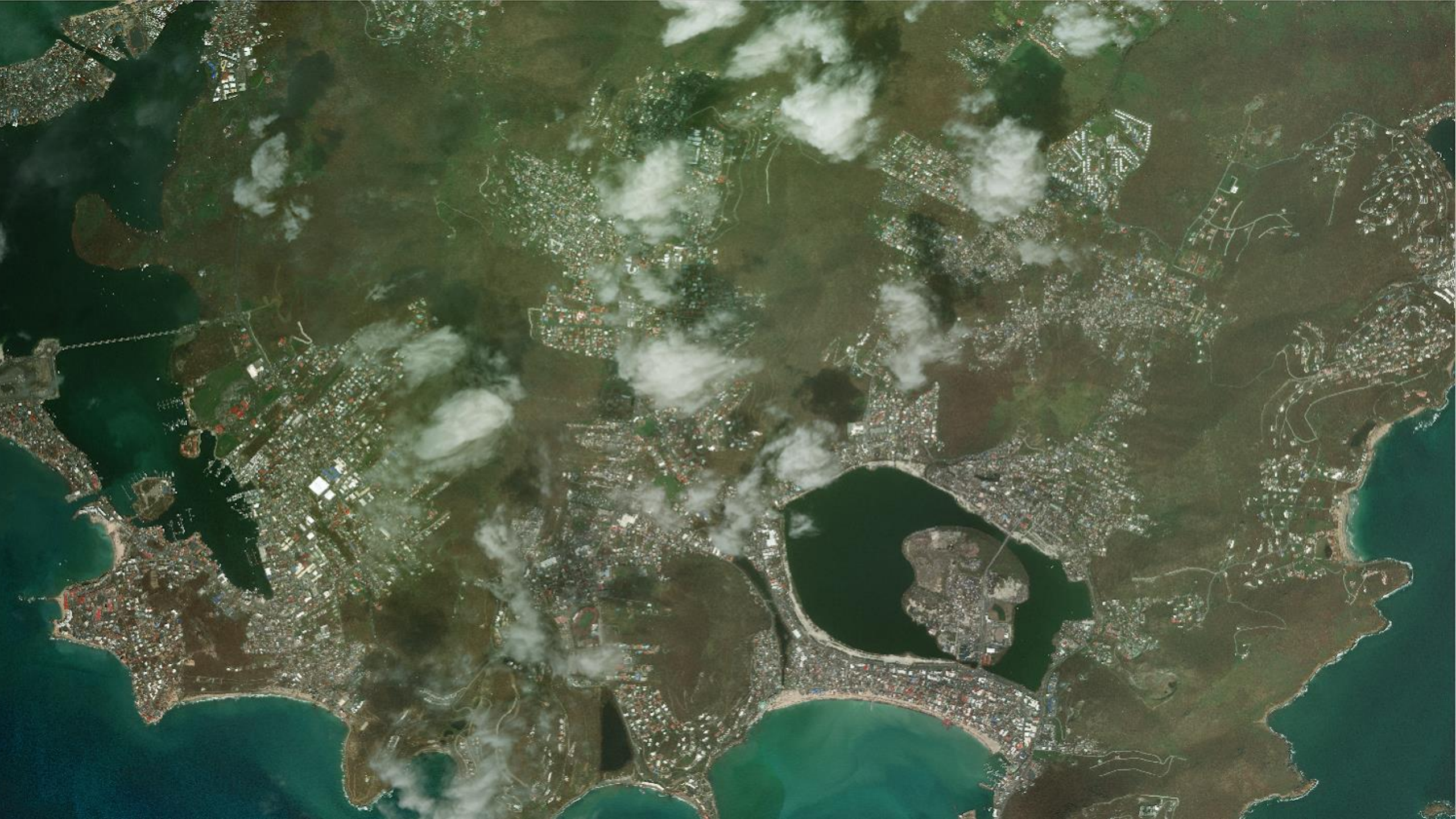
HOW WE DO IT





AFTER IRMA:
DRONE IMAGERY
SOURCE: 510
BLUE AREAS TARPAPULINE

510   Red Cross  St. Maarten





Emergency responders *immediately* need to know:

- **WHERE** are the people in need
- **HOW BAD** is the situation → the scale of the damage

How can we do it?

- By going to the field and surveying
 - often impossible, takes weeks/months for large areas
- By manually checking satellite imagery
 - too slow if affected area is large

Emergency responders *immediately* need to know:

- **WHERE** are the people in need
- **HOW BAD** is the situation → the scale of the damage

How can we do it?

- By going to the field and surveying
 - often impossible, takes weeks/months for large areas
- By manually checking satellite imagery
 - too slow if affected area is large

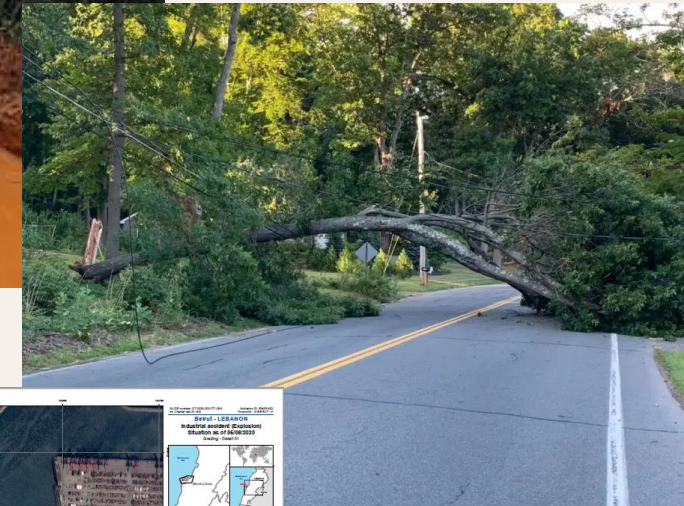


Emergency responders *immediately* need to know:

- **WHERE** are the people in need
- **HOW BAD** is the situation → the scale of the damage

How can we do it?

- By going to the field and surveying
 - often impossible, takes weeks/months for large areas
- By manually checking satellite imagery
 - too slow if affected area is large



Automation is about **speed** and **availability**

Mapping:

- one machine: 1,000 km²/h *
- HOT OSM (many humans): < 100 km²/h **

Damage Assessment (on satellite images):

- one machine: 10,000 buildings/h *
- UNOSAT: < 100 buildings/h/person ***

* Azure's NC24: 24 vCPUs, 224 GB RAM, 4x NVIDIA K80 GPUs, 3.60 USD/hour

** <https://doi.org/10.1080/22797254.2018.1460567>

*** <https://www.mdpi.com/2072-4292/12/17/2839>

Automation is about **speed** and **availability**

Mapping:

- **one** machine: 1,000 km²/h *
- HOT OSM (many humans): < 100 km²/h **

Damage Assessment (on satellite images):

- one machine: 10,000 buildings/h *
- UNOSAT: < 100 buildings/h/person ***

* Azure's NC24: 24 vCPUs, 224 GB RAM, 4x NVIDIA K80 GPUs, 3.60 USD/hour

** <https://doi.org/10.1080/22797254.2018.1460567>

*** <https://www.mdpi.com/2072-4292/12/17/2839>

Automation is about **speed** and **availability**

Mapping:

- **one** machine: 1,000 km²/h *
- HOT OSM (many humans): < 100 km²/h **

Damage Assessment:

- **one** machine: 10,000 buildings/h *
- UNOSAT: < 100 buildings/h/person ***

* Azure's NC24: 24 vCPUs, 224 GB RAM, 4x NVIDIA K80 GPUs, 3.60 USD/hour

** <https://doi.org/10.1080/22797254.2018.1460567>

*** <https://www.mdpi.com/2072-4292/12/17/2839>

Automation is about speed and availability

Mapping:

- **one** machine: 1,000 km²/h *
- HOT OSM (many humans): < 100 km²/h **

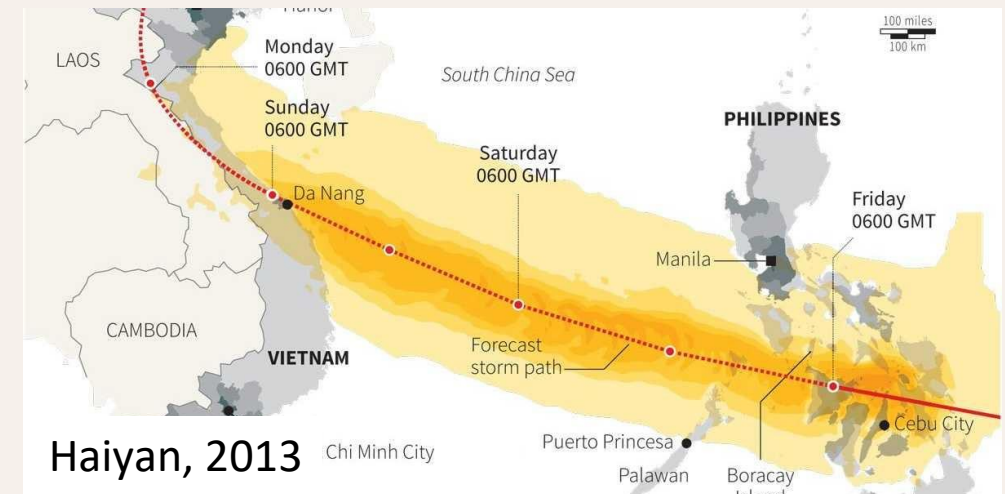
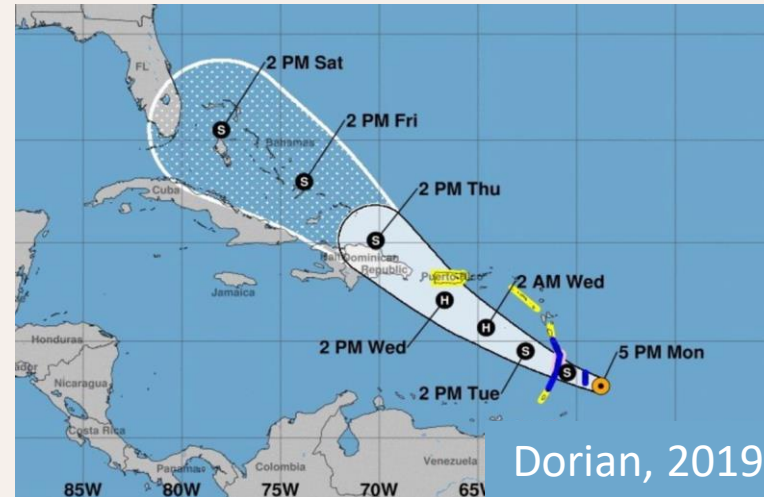
Damage Assessment:

- **one** machine: 10,000 buildings/h *
- UNOSAT: < 100 buildings/h/person ***

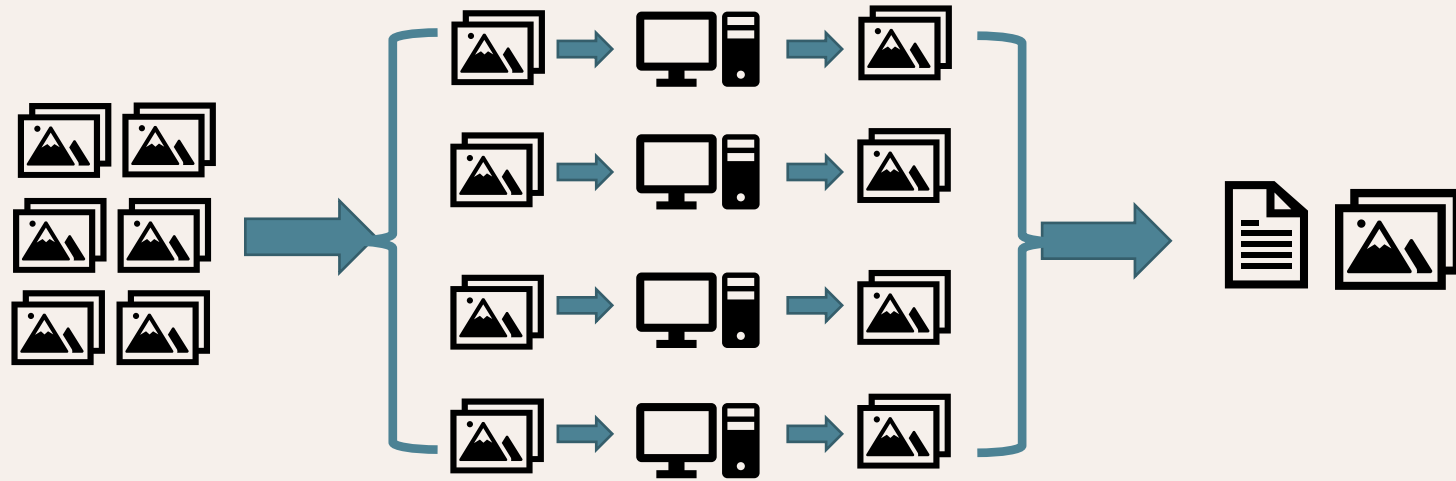
* Azure's NC24: 24 vCPUs, 224 GB RAM, 4x NVIDIA K80 GPUs, 3.60 USD/hour

** <https://doi.org/10.1080/22797254.2018.1460567>

*** <https://www.mdpi.com/2072-4292/12/17/2839>



- Machines can also work in parallel → will go **as fast as you can pay for**

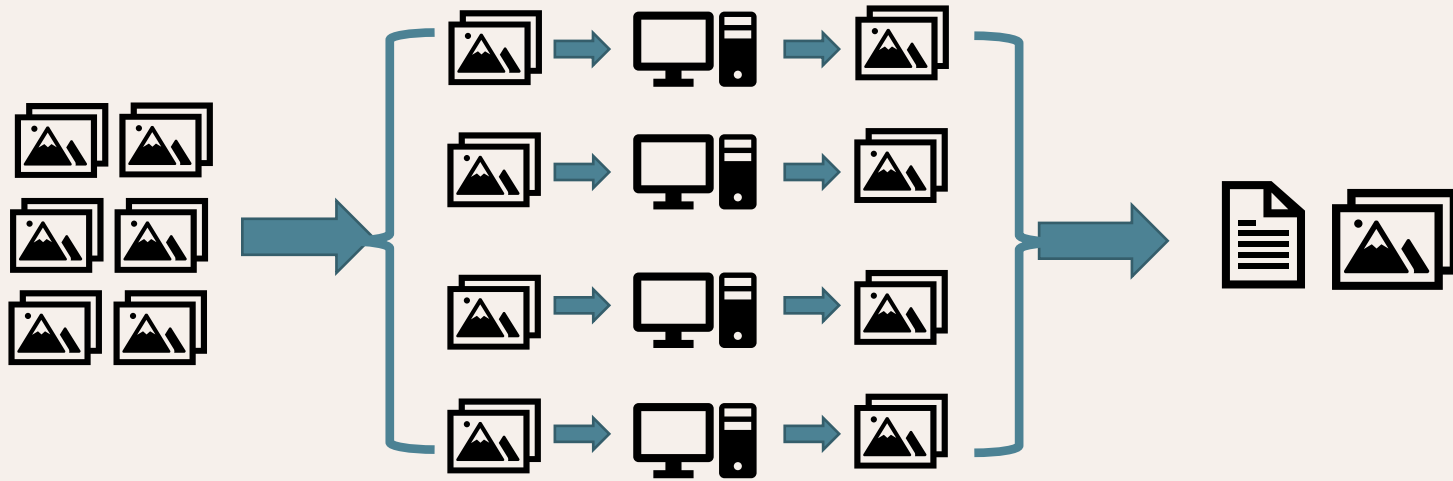


github.com/rodekruis/ada-azure-batch

- Example: mapping + damage on ~12,000 km²* on Azure with 26 x NC12 VMs:
completed in 1.2 h for 24 EUR

* Super typhoon mangkhut, Philippines, 2018

- Machines can also work in parallel → will go **as fast as you can pay for**



github.com/rodekruis/ada-azure-batch

- Example: mapping + damage on ~12,000 km²* on Azure with 26 machines (NC12 promo):
completed in 1.2 h, cost 24 EUR

* Super typhoon mangkhut, Philippines, 2018

OUTLINE

WHY AUTOMATED DAMAGE ASSESSMENT

HOW WE DO IT

STEP 1: BUILDING DETECTION

input: satellite image (RGB, ≤ 0.5 m/pixel)



output: building locations (vector)



source: Bing Maps (fully automated) or manual input

STEP 2: BUILDING DAMAGE CLASSIFICATION

1. satellite images, pre- and post-disaster (RGB, 0.5 m/pixel)
2. building locations (vector)



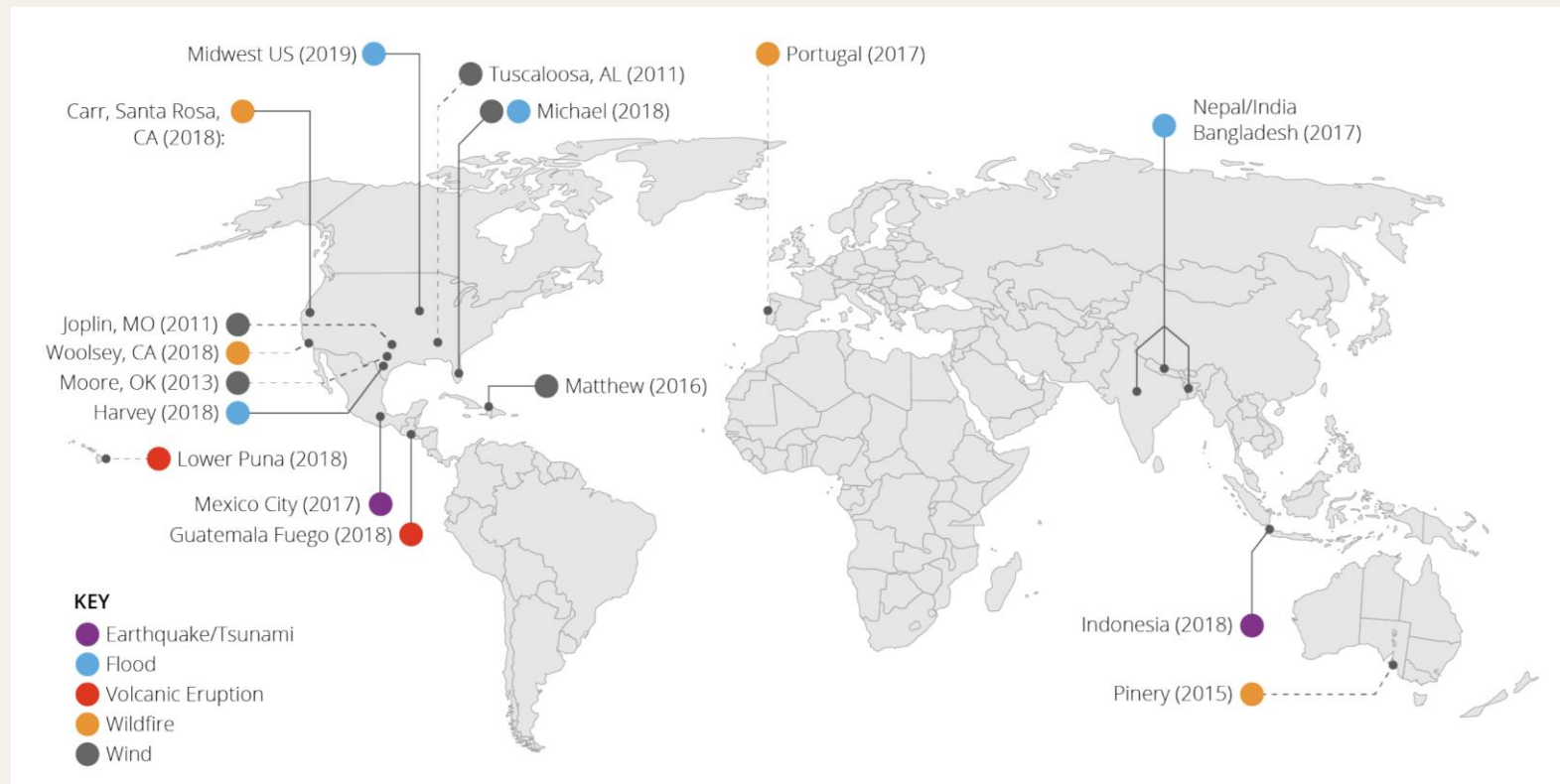
output: building damage (vector), 4 classes



source(s): Maxar Open Data or manual input

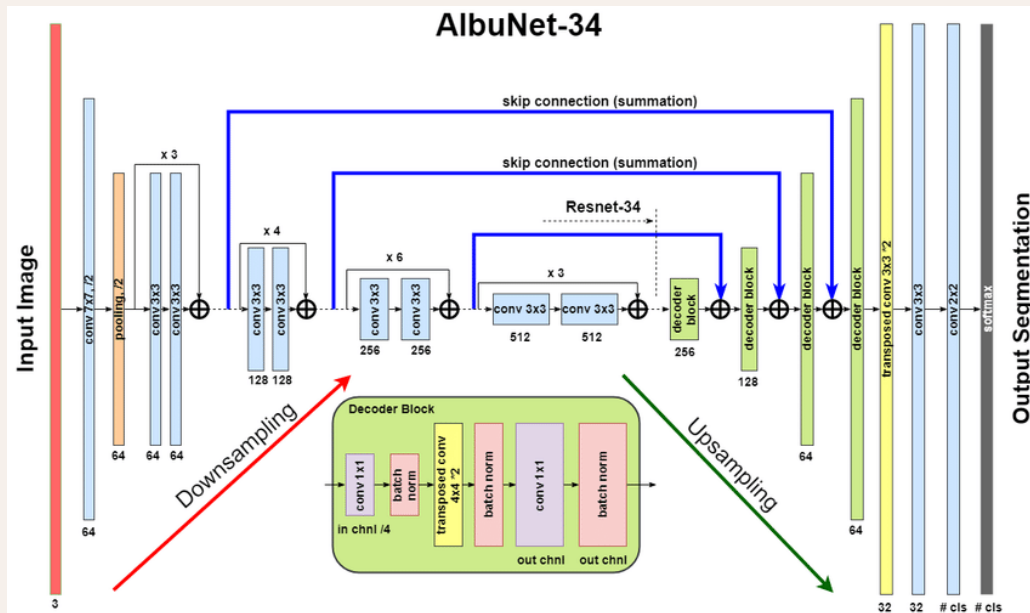
Training data: **xBD dataset** ([arXiv:1911.09296](https://arxiv.org/abs/1911.09296))

850k buildings, 45k km², 4 continents, 19 disasters

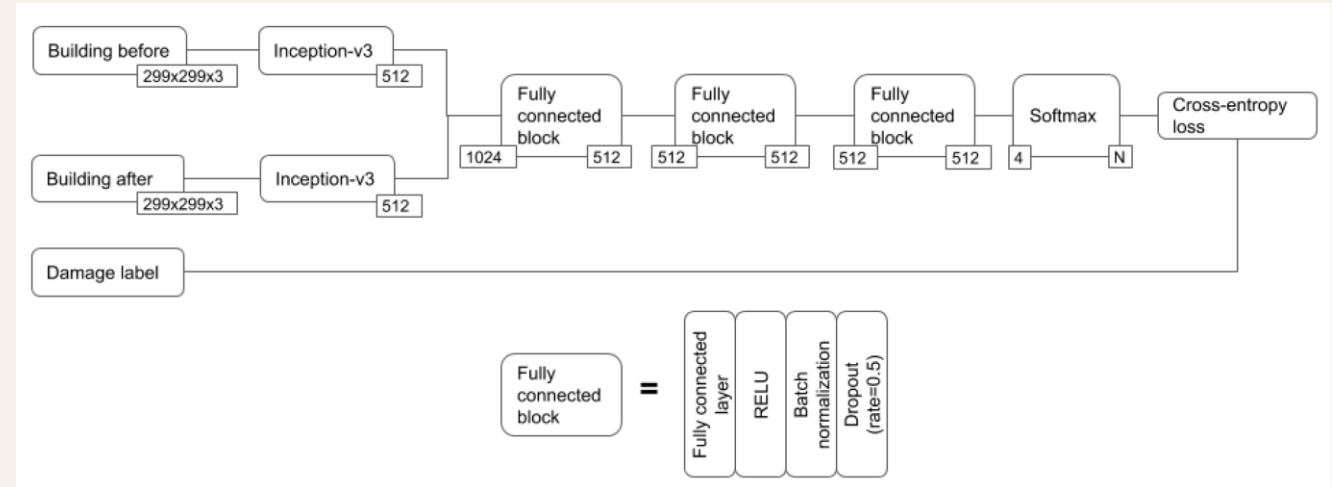


Model architectures:

- Building detection (segmentation): U-Net with ResNet encoder (a.k.a. AlbuNet)
- Building damage classification: pseudo-siamese Net



[arXiv:1804.08024v1](https://arxiv.org/abs/1804.08024v1)

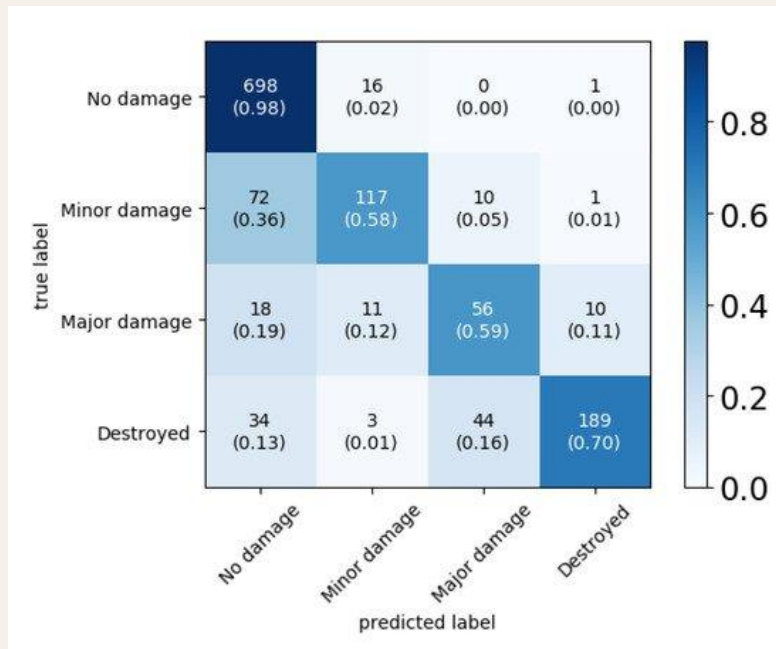


[doi:10.3390/rs12172839](https://doi.org/10.3390/rs12172839)



Key performance indicators:

- Building detection (segmentation): IoU 0.71, MCC 0.68
- Building damage classification, Macro F1 (4 classes): 0.59-0.79



Key **performance** indicators:

- Building detection (segmentation): IoU 0.71, MCC 0.68
- Building damage classification, Macro F1 (4 classes): 0.59-0.79

Additional observations ([doi:10.3390/rs12172839](https://doi.org/10.3390/rs12172839)) :

- Little (<10%) effect on performance of
 - world region (continent)
 - damage type (water, wind, etc.)
 - image parameters (off-nadir angle, panchromatic resolution, etc.)
- Performance strongly influenced by **which specific disaster** models are trained and tested on

- **Automated Mapping and Damage Assessment** can be useful tools in Emergency Response
 - Faster and more often available than current methods

- **Automated Mapping and Damage Assessment** can be useful tools in Emergency Response
 - Faster and more often available than current methods
- Model performance tested on a number of disasters in different world areas
 - Good enough for first estimates, more work to be done

- **Automated Mapping and Damage Assessment** can be useful tools in Emergency Response
 - Faster and more often available than current methods
- Model performance tested on a number of disasters in different world areas
 - Good enough for first estimates, more work to be done
- Models and framework are **fully open source**, so that the community can use & improve them
 - Automated mapping: <https://github.com/rodekruis/automated-building-detection>
 - Automated Damage Assessment: <https://github.com/rodekruis/caladrius>

- Automated Mapping:
 - Typhoon Goni, Philippines 2020
 - Hurricane Eta and Iota, Guatemala 2020
 - Flash Floods, Zimbabwe 2020
- Damage Assessment:
 - Explosion, Beirut 2020

The screenshot shows the HDX (Humanitarian Data Exchange) interface. At the top, there's a navigation bar with 'OCHA Services', 'Data Responsibility for COVID-19', 'FAQ', 'Log in', 'Sign up', and 'Switch to HDX Lite'. Below this is a search bar and navigation links for 'DATA', 'LOCATIONS', 'ORGANISATIONS', and 'QUICKLINKS'. A red 'ADD DATA' button is on the right. The main content area is titled 'AI building footprint in southern Guatemala' and features the Rode Kruis logo. The description states: 'Southern Guatemala: AI predictions of building footprint on Bing Maps images (approximately 2016-2019), see <https://github.com/rodekruis/automated-building-detection>. Produced in support to DRRT Guatemala for hurricane Eta and Iota. Coordinate reference system: WGS 84 / EPSG:4326'. It also shows '30+ Downloads' and 'This dataset updates: Never'. A 'Contact the contributor' link and social media icons are present. The main part of the image is a map of southern Guatemala with a green overlay representing the AI-predicted building footprint. A 'Shape info' tooltip is visible over a shape on the map. The map includes labels for various cities and regions like Huehuetenango, Santa Cruz del Quiché, Baja Verapaz, Zacapa, Chiquimula, El Progreso, Jalapa, Ciudad de Guatemala, Antigua Guatemala, Escuintla, Ahuachapán, Suchitoto, Ilobasco, Chimaltenango, Nueva Ocotepeque, Santa Rosa de Copán, Santa Bárbara, Siguatepeque, Comayagua, La Esperanza, La Paz, Jocaitique, and Ahuachapán. The map is powered by Leaflet and OpenStreetMap contributors.

<https://data.humdata.org/organization/netherlands-red-cross>

THANK YOU!

contact: jmargutti@redcross.nl