PUBLICATIONS · ACADEMIC REPORT

HASTE: HIGH-SPEED ASSESSMENT AND SATELLITE TRACKING FOR EMERGENCIES

RAPID POST-DISASTER BUILDING DAMAGE ASSESSMENT

Software by the Microsoft AI for Good Lab · Report by Ethical Tech CoLab
NYU Center for Global Affairs
July 2026

HASTE was developed by the Microsoft AI for Good Lab and released as open-source software under the MIT Licence. The contributors recorded in the repository's commit history are Meygha Machado, Caleb Robinson, Joaquín Rivero, Cameron Birge, Marcelo Duarte, and Anthony Cintron Roman. The accompanying research paper additionally credits Anthony Ortiz, Simone Fobi Nsutezo, Kevin White, Inbal Becker-Reshef, and Juan M. Lavista Ferres. This plain-language report was prepared under the Ethical Tech CoLab at the NYU Center for Global Affairs as part of masters research (2026), on the fork of the project held at Ethical-Tech-CoLab/haste.

> HASTE is an open-source platform that lets a trained analyst, working without code, fit a disposable machine-learning model to a single disaster and produce a building-by-building damage estimate within hours. This report reads the platform's own source code alongside its documentation to set out what every adjustable and hardcoded setting means, what the published performance figures do and do not establish, and where the answer it produces is constrained by things the software does not control.

KEY FIGURES

31

field deployments since early 2023, developer-reported (§08)

3

labelled buildings, across at least two categories, before the in-browser classifier will train

0.1

damage threshold, one tenth of a building's visible area, the most consequential single number in the platform (§05)

0.84

area under the ROC curve at one per cent of labels, against 0.88 fully supervised (§08)

00

FOREWORD

In the hours after an earthquake, a hurricane, or a wildfire, the single most useful thing a relief coordinator can hold is a map of which buildings are still standing. Where should search teams go first? Which neighbourhoods need shelter, and how many people are likely to need it? Which roads lead to places that no longer exist? Those questions have to be answered before anyone on the ground has been able to walk the affected area, and often before the affected area is safe to walk at all.

The traditional answers are slow. Teams fly helicopters over the damage, photograph it, and interpret the photographs by hand. Official mapping services task a satellite, wait for a cloud-free pass, and publish a product days later. Both are careful and both are valuable, but neither reliably delivers inside the first days, which is the window in which decisions about people, supplies, and attention are actually being made.

HASTE, which stands for High-speed Assessment and Satellite Tracking for Emergencies, is an open-source research platform built to shorten that gap. It lets a trained analyst who is not a machine-learning engineer take fresh satellite or aerial imagery of a disaster zone, mark a small number of examples by hand, and have a computer extend those examples across the whole affected area to produce a building-by-building damage estimate. This report

explains, in non-technical language, what HASTE does, how it does it, what each of its settings means, and what it cannot be trusted to do.

01

EXECUTIVE SUMMARY

HASTE is a web-based platform that turns post-disaster satellite imagery into an estimate of which individual buildings have been damaged. It was developed by the Microsoft AI for Good Lab and released as open-source software under the MIT licence. The version reviewed here is the copy held in the Ethical Tech CoLab repository, which is a fork of the original project at microsoft/haste.

The platform's central design choice is to train a fresh model for each disaster rather than maintain one global model that tries to recognise damage everywhere. A hurricane in Jamaica and an earthquake in Türkiye leave visually different traces on visually different building stock, and a model tuned to one is not expected to work on the other. HASTE accepts that limitation deliberately in exchange for speed: a model fitted to one event, from one analyst's labels, can be ready in minutes.

A human being is required at every stage, and there is no automatic mode. The project documentation states repeatedly that outputs are preliminary signals requiring expert validation rather than authoritative damage assessments. Section 09 sets out where the human sits in the workflow.

HASTE offers two routes from imagery to an answer. The faster route, Rapid Building Assessment, computes a numerical fingerprint for every building in the area, asks the analyst to label a handful of them, and trains a very small classifier inside the web browser, which scores all the rest in seconds. The slower route, Damage Mapping, asks the analyst to draw damaged and undamaged areas by hand and trains a full image-segmentation model on a graphics processor, producing a continuous, pixel-level damage map.

Both routes end at the same place: a per-building damage figure, a set of accuracy measures computed against a human-labelled validation sample, and an estimate of the total number of damaged buildings with a stated margin of error.

The platform depends on outside data it does not produce. Imagery comes from commercial and public providers such as Planet, Maxar, Airbus, the European

Union's Copernicus programme, and the United States National Oceanic and Atmospheric Administration. Building outlines come from the Overture Maps Foundation. Where those outlines are missing or wrong, HASTE has nothing to attach its predictions to.

According to the research paper published alongside the platform, HASTE has been used in thirty-one field deployments since early 2023, and its outputs have been released openly through the Humanitarian Data Exchange.

02

BACKGROUND AND RATIONALE

The problem. Rapid damage assessment is a timing problem before it is a technical one. Imagery of a disaster zone often becomes available within a day. An interpretation of that imagery detailed enough to direct a response team to a particular neighbourhood usually does not. The interval between the two is where humanitarian decisions are made with the least information and the greatest consequences.

The gap. Established products have real strengths and known constraints. Copernicus Emergency Management Service Rapid Mapping, the European Union's free on-demand crisis mapping service, delivers standardised map products within hours to days of an activation, but must be formally activated by an authorised user and covers only large-scale emergencies. Manual aerial surveys are accurate but limited in geographic reach and slow to process. Neither easily absorbs the specific situational context that a particular responding organisation cares about, such as one parish, one road corridor, or one category of structure.

The nearer comparator is the automated damage classification literature that grew out of the xView2 challenge, on whose xBD dataset HASTE is itself benchmarked. Against that work the claim is not that nothing existed. It is that existing machine learning approaches assume a globally pretrained model applied to a new disaster, and HASTE trades that for a disposable per event model an analyst fits by hand, in a browser, without writing code.

The response. The proposition HASTE was built around, carried over from earlier in-browser damage-assessment research at the same laboratory, is that human oversight should be structural rather than advisory: the operator is not reviewing a machine's conclusion after the fact, the operator is the source of everything the machine knows about this event.

The design also reflects a practical constraint on who does this work. The people who understand what damage looks like in a given country are rarely the people who can write machine-learning code. HASTE is presented as a no-code platform so that the person supplying the expert judgment and the person operating the model can be the same person.

03

OBJECTIVES

The platform is designed to do the following.

- Produce a building-level damage estimate from post-disaster imagery quickly enough to be useful inside the first days of a response.
- Allow a non-programmer to fit a model to a specific event using only a map interface, a mouse, and their own visual judgment.
- Keep a human in control of every consequential step, including the decision to release a result at all.
- Express uncertainty honestly, by measuring the model against a separate human-labelled sample and reporting a margin of error on the headline damage figure rather than a single confident number.
- Produce outputs in standard geographic file formats so that they can be opened in the mapping software humanitarian organisations already use, and published openly where appropriate.
- Remain deployable by others. The platform can be run on a laptop for evaluation or installed on an organisation's own cloud infrastructure, in which case that organisation, and not Microsoft, controls the imagery and the outputs.

04

HOW HASTE WORKS

The workflow has a shared beginning, then splits into two routes, then rejoins for validation and reporting.

The shared beginning. An analyst first creates a project, which is simply a container for one disaster event. It records a name, a description, the date of the event, and the affected countries. The date and the countries are stored for reference and do not affect any calculation.

Into that project the analyst adds an image layer, which is the imagery being assessed. It arrives as GeoTIFF, a standard image format that carries the geographic coordinates of every pixel alongside the picture itself. The analyst can upload several files covering the same area, and HASTE merges them into a single mosaic. Imagery from before the event is optional and is used for visual comparison, not for any calculation.

HASTE then obtains the outlines of every building in the area covered by the imagery. By default it downloads them automatically from Overture Maps, an open dataset maintained by the Overture Maps Foundation under the Linux Foundation, which combines OpenStreetMap with machine-derived building datasets from Microsoft and Google and, on the foundation's own account, covers roughly 2.3 billion buildings worldwide. An analyst who has better local data can upload their own outlines instead, as a GeoPackage file of up to 500 megabytes; HASTE converts it to the standard global coordinate system and trims it to the imagery area. When the layer is created the analyst chooses a workflow, Building or Standard, which determines which of the two routes is available.

Route A, Rapid Building Assessment. This route is available when building outlines exist and produces an answer in minutes with no separate training job. First, HASTE computes an embedding for every building. An embedding is a compact list of numbers that describes what the imagery around that building looks like: its texture, its colour, its edges. Two buildings that look alike receive similar lists of numbers. Nothing in the embedding knows anything about damage; it is simply a numerical description of appearance.

The analyst then opens a map, clicks buildings, and assigns each one to one of three categories: Intact, Damaged, or Cloudy, the last meaning that cloud or shadow makes the building impossible to judge. A left click labels a building, a right click removes the label, and holding the control key while dragging labels many at once.

Once at least three buildings have been labelled across at least two categories, a very small statistical model called a logistic regression trains automatically in the browser and immediately predicts a category for every other building in view. The analyst can flip between what they labelled and what the model predicted, see where it is wrong, label a few more examples, and watch the prediction improve. This loop takes seconds, which is what makes the route fast. When satisfied, the analyst runs a final pass that scores every building in the layer and saves the result as a geographic data file.

Route B, Damage Mapping with a trained model. This route does not require building outlines and produces a continuous damage picture across every pixel of the imagery rather than a verdict per building. The analyst draws polygons, rectangles, or circles over a small part of the image and assigns each shape a class. The default classes are Background, Building, and Damaged Building; additional classes of No Damage and Flood Extent are offered for particular event types. The project documentation advises drawing at least five to ten shapes per class, considers seventy to one hundred a good training set, and states that more than roughly one hundred and fifty is unnecessary.

Those shapes train an image-segmentation model, which is a model that assigns a class to every individual pixel rather than to a whole picture. The architecture used is a U-Net with a ResNeXt-50 encoder, a standard and well-understood design in satellite image analysis, started from weights pre-trained on ordinary photographs and then fitted to the analyst's labels. Training runs on a graphics processor, either locally in a container or on cloud computing capacity. The trained model is then run across the entire image layer, producing a per-pixel damage prediction that can be viewed alongside the imagery and downloaded.

Turning pixels into buildings. A per-pixel damage map is not directly useful to a responder, who needs to know about buildings. HASTE therefore overlays the building outlines onto the pixel predictions and, for each building, counts what proportion of the classified pixels inside that outline were called damaged. That proportion, between zero and one, is the building's damage fraction. It is the number that drives everything downstream. The same step also records what proportion of each building was obscured by cloud.

Validation. Both routes end with the same discipline. The analyst opens a validation tool that presents a random sample of roughly two hundred building outlines over the post-event imagery and asks them to judge each one independently as Damaged, Not Damaged, or Unknown. These human judgments are treated as the ground truth against which the model's predictions are scored. Buildings marked Unknown are excluded from the scoring entirely rather than being guessed at.

THE VARIABLES, EXPLAINED SIMPLY

Every number an analyst can adjust, and every number fixed inside the code, is described below in ordinary terms: what it represents, why it is set where it is, and what it changes about the answer.

The label classes. These are the categories an analyst draws with. Each is stored internally as a class value, and later steps identify a class by that position rather than by its name.

CLASS	STORED VALUE	WHAT IT MARKS
Background	1	Ground and anything that is not a building.
Building	2	An intact structure.
Damaged Building	3	A structure showing damage. This is the class the damage figures are computed from.
Cloud	4	Areas where cloud or shadow makes the imagery impossible to read.

The four positional label classes and the values they are stored as.

Two of these choices carry more weight than they appear to. The reason Background exists as a class in its own right is that the model is only taught by the pixels the analyst actually marked, so if a damaged roof is labelled without the ground around it, the model is never told what the ground is, and can produce a blurry, spreading prediction without ever being penalised for it. Labelling a building together with its surroundings is what forces the model to learn a boundary. And cloudy buildings are excluded from the damage statistics rather than counted as intact, which would otherwise bias the result downward in exactly the wet, storm-affected conditions where damage assessment matters most.

No Damage and Flood Extent. Two additional classes offered for earthquake, fire, and flood event types respectively. Flood Extent lets the analyst mark standing water, which HASTE can intersect with building outlines to say which buildings are in water.

Normalisation means and standard deviations. Image pixels are stored as whole numbers, but models learn more stably from values between 0 and 1, so

each channel is rescaled by subtracting a mean and dividing by a standard deviation. In the worked example shipped with the platform these are set to 0 and 255, which is a plain rescaling of ordinary 8-bit imagery. When an image layer is prepared automatically, HASTE instead sets every mean to zero and sets each channel's divisor to the brightness value at that channel's 98th percentile, so that the brightest two per cent of pixels are treated as glare and clipped rather than being allowed to compress everything else into a narrow band. This matters because imagery from different sensors arrives on different numeric scales, and a wrong divisor makes the picture unrecognisable to the model.

Ground resolution. HASTE never checks, records, or standardises how many metres of ground each pixel covers. It works on whatever grid the supplied imagery has. Several settings are expressed in pixels rather than metres as a result, and the physical area covered by a training tile therefore varies with the imagery source.

Learning rate, default 0.0001. How large a correction the model makes each time it discovers it was wrong. Too large and it lurches past the right answer; too small and it never gets there within the time available. The default is a conservative value appropriate to fine-tuning a model that already carries useful pre-trained weights.

Batch size, default 32 for training. How many image tiles the model examines before making one correction. Larger batches give a steadier, less noisy signal about which direction to move in, but require proportionally more memory on the graphics processor.

Maximum epochs. One epoch is one complete pass through the training data. A small number is intended here, for a clear reason: the model is being fitted to a few dozen hand-drawn shapes from a single event, and training it longer would mainly teach it to memorise those particular shapes rather than the general appearance of damage. The repository is inconsistent about the actual figure. The worked example specifies 10, the form in the web interface offers 3, and the server falls back to 1 when nothing is supplied, while the training script separately imposes a hardcoded minimum of 10. An analyst who selects 3 in the interface is therefore asking for a maximum below the enforced minimum, which anyone reproducing a result should know.

Training chip size, 256 pixels. The model does not see the whole scene at once. It is shown small square cut-outs 256 pixels on a side, drawn at random from the labelled area, and the training run feeds it 1,024 batches

of them per epoch. The model's whole view of the disaster is assembled from these fragments.

Label buffer, nominally 3 metres. Applied to the Building class using the Background class. When a person traces a building, the traced line is never exactly on the wall. This setting draws a narrow band around each building polygon and treats it as Background, which stops the imprecision of human tracing from teaching the model that the pavement is part of the structure. The setting is written in metres but is applied by counting pixels, so it means three metres only when a pixel happens to represent one metre of ground. The script's own documentation notes this.

Patch size, default 2048, and padding, default 64. Used when running the finished model across the full image. Satellite scenes are far too large to process in one piece, so they are cut into square tiles. Predictions are least reliable at the very edge of a tile, where the model can see no context, so each tile is processed with 64 pixels of overlap that are then discarded. This is what stops a visible grid pattern from appearing across the finished damage map.

Initial weights, default none. Meaning the model starts from weights learned on ordinary photographs. An analyst can instead start from a previously trained HASTE model held in the model catalogue, which is offered only where the event type and imagery source match, since a model fitted to wildfire imagery is not a sensible starting point for flood imagery.

The cloud penalty. An optional setting, switched off by default, changes how the model is corrected in areas the analyst marked as Cloud. Rather than being told what class those pixels really are, which nobody knows, the model is penalised whenever it predicts damage beneath a cloud. In plain terms it is taught not to claim to see through weather.

Embedding backbone. The method used to convert the appearance of a building into numbers. The lightweight default is MOSAIKS, an approach published in Nature Communications in 2021 that passes the image through a large set of randomly generated filters rather than learned ones. The counter-intuitive finding of that work is that random filters, applied at sufficient scale, describe an image well enough for a simple linear model to make accurate predictions from them, at a tiny fraction of the computational cost. The alternative offered is DINOv2, a family of vision models released by Meta AI in 2023 that learned general visual features from 142 million unlabelled images, available in small and base sizes.

Output dimensions, default 1024. Applicable to MOSAIKS only. How many numbers describe each building. More numbers capture more nuance and cost more time and memory. When DINOv2 is used this setting is ignored, because the size of the description is fixed by the model itself: 384 numbers for the small version, 768 for the base version, 1024 for the large one.

Kernel size, default 7. The width in pixels of each random filter. Small filters respond to fine texture such as roof material and rubble; larger ones respond to coarser structure. Seven pixels is a middle setting suited to detecting the change in surface texture that collapse produces.

Resize factor, default 4. How much the small image crop around each building is enlarged before being described. In moderate-resolution satellite imagery a single house may span only a handful of pixels, too few for the filters to find anything. Enlarging the crop fourfold gives them something to work with. This is a real trade-off and not a free gain: enlarging an image does not create detail that the sensor never recorded, it only makes the recorded detail large enough to be measured. Choosing DINOv2 forces this setting back to 1, because that model brings its own fixed way of dividing an image into pieces.

Context padding, 8 pixels. Each building crop reaches a little beyond the traced outline, so that the description includes some of the ground immediately around the structure. Debris and scorch marks sit there rather than on the roof.

What is actually described. The crop is divided into a grid of small tiles, each tile is described separately, and only those tiles falling inside the building's outline are averaged together to produce the building's final list of numbers. Buildings that fall outside the imagery keep their place in the file with an empty entry rather than being dropped, because everything downstream matches buildings to predictions by position in the file rather than by name. That design decision is efficient and fragile in equal measure.

The crop cap, 192 source pixels. A limit that stops a single very large building, such as a warehouse or a stadium, from generating an enormous crop and exhausting the available memory. Oversized footprints are cropped from the centre instead.

The in-browser classifier. The model that turns a handful of labelled buildings into predictions for all of them is a logistic regression, one of the oldest and simplest classifiers in statistics. It fits a straight-line

boundary through the space of building descriptions and reports, for each building, how far onto the damaged side of that boundary it falls. Its learning rate is 0.1 and it takes 500 steps, values fixed in the code rather than exposed to the analyst, chosen to complete in well under a second so that the label-and-see loop stays interactive.

Regularisation strength, 0.01. A deliberate penalty that discourages the model from placing heavy reliance on any single one of the thousand-odd numbers describing each building. Without it, a model trained on three examples would seize on whatever coincidental feature happened to separate those three, and would fail on the fourth. This single small number is the main defence against overfitting in a workflow explicitly built around very few labels.

Holdout fraction, 0.2 in practice. One fifth of the analyst's labels are held back from training and used only to score the model, which is what produces the live precision, recall, and F1 figures shown for the Damaged class in the side panel. Because those numbers come from labels the model was never shown, they are an honest indication of quality rather than a report of how well the model memorised its own training data. Two weaknesses are worth naming. The sample is tiny: with twenty labels the holdout is four buildings, and four buildings cannot tell you much. And the split is drawn afresh every time a label is added, so the displayed figures jump around from click to click, as the code comments acknowledge.

Measurement rings, 0, 10, and 20 metres. Not to be confused with the label buffer above. HASTE calculates each building's damage fraction three times: once inside the traced outline, once inside a ring extending ten metres beyond it, and once at twenty metres. There are two reasons. Building outlines and satellite imagery are frequently misaligned by several metres, particularly in dense cities and where the ground itself has moved, so the strict outline may sit over the neighbouring plot. And some kinds of damage, notably debris fields and burn scars, appear around a structure rather than on it. The wider measurements are recorded so that this can be examined rather than assumed.

Damage fraction. For a given buffer, the number of pixels classified as Damaged Building divided by the number of classified pixels of any kind inside that shape. Unclassified pixels, recorded as zero, are excluded from both the top and the bottom of that division, so a building half outside the imagery is judged on the half that was actually seen. The result is capped at 1.

Damage threshold, default 0.1. The proportion of damaged pixels above which a building is declared damaged. This is the most consequential single number in the entire platform, and it is set low on purpose. A tenth of a building's visible area showing damage indicators is treated as enough, reflecting a judgment that in the first days of a response, missing a damaged building is a worse error than flagging one that turns out to be intact. It also compensates for partial visibility: a structure photographed from overhead shows mainly its roof, and serious structural damage may present as only a modest patch of altered texture there. Because the threshold is a setting rather than a fixed rule, and because the platform also publishes a full precision-and-recall curve across all possible thresholds, an analyst who disagrees with this trade-off can see exactly what a different choice would cost.

A wrinkle a careful reader should know about. The exported data file also carries a simple yes-or-no damaged column, and that column is set whenever a building contains even one damaged pixel, with no threshold at all. The validation report reads that column; the assessment report applies the tenth-part threshold. The same model and the same human labels will therefore yield slightly different accuracy figures in the two reports. Nothing in the documentation flags this, and it is the kind of discrepancy that could confuse a reader comparing the two.

Minimum footprint area, default 50 square metres. Buildings smaller than this are excluded from the population used for the final headline estimate. The stated reasoning is that these are structures a human reviewer could not realistically have judged in the validation step, so including them in an extrapolation built from human judgments would be unsound. The consequence, which the platform does not hide, is that small informal structures are absent from the headline count, and those structures are common in precisely the settlements most exposed to disaster.

Cloud exclusion. Any building with a cloud fraction above zero is excluded from the damage count entirely. This is a strict rule, not a proportional one; a building need only be slightly obscured to be set aside.

The final estimate. HASTE does not publish a raw count of the buildings its model called damaged. It uses the human-validated sample to estimate the true damage rate and scales that rate up to the full population of buildings. The arithmetic is a standard finite-population survey estimate. The sample proportion, written as p-hat, is the number of buildings a human confirmed as damaged divided by the number of buildings the human judged either way, with Unknown buildings excluded from both figures. The

population, written as N, is the count of building outlines larger than the minimum area, and the estimated number of damaged buildings is simply N multiplied by p-hat.

The sampling fraction, written as f. The size of the validated sample divided by the population. It enters the calculation through what statisticians call a finite-population correction, the term (1 minus f) in the variance. Its effect is intuitive: if you have checked half of all the buildings by hand, your uncertainty about the other half should be smaller than if you had checked one per cent of them. Standard survey formulas assume an infinite population and would overstate the uncertainty here.

The critical value z, fixed at 1.959963984540054. This is the constant that turns a standard error into a 95 per cent confidence interval, and it is written out as a literal number in the code specifically so that the platform does not have to load a heavy statistical library to obtain it. The reported interval runs from N times (p-hat minus z times the standard error) to N times (p-hat plus z times the standard error).

What the interval means in plain terms. If the same sampling exercise were repeated many times, an interval calculated this way would contain the true number of damaged buildings in about nineteen cases out of twenty. It accounts for the fact that a sample was taken. It does not account for a mislabelled sample, misaligned building outlines, or a model that is wrong in a consistent direction, and it should not be read as expressing total uncertainty about the figure.

06

READING THE RESULTS

The visualiser. The pre-event and post-event imagery are shown side by side with a swipe control between them, the predicted damage layer overlaid on both, and the raw per-pixel predictions available as an optional toggle. Opacity, contrast, hue, and saturation sliders help make damage visible in imagery that was captured in poor light.

On the map, each building is shaded according to its damage fraction in five bands, cut at one fifth, two fifths, three fifths, and four fifths, running from white through pale peach and orange to red and dark red. This is a presentational scale only. It should not be read as a severity classification in the sense used by structural engineers, because the model

was never taught degrees of damage. It only ever learned to separate damaged from intact, and the shading reflects how much of a roof it flagged rather than how badly the building was hurt.

The validation report. This compares the model's predictions against the analyst's own validation labels and reports overall accuracy, precision, recall, and F1 for each class, a macro-averaged F1, and a confusion matrix. Precision answers the question: of the buildings the model called damaged, what share really were? Recall answers the opposite: of the buildings that really were damaged, what share did the model find? These two errors have different humanitarian costs, and the report deliberately does not merge them into a single figure.

The assessment report. This summarises the whole layer: the total number of building outlines, how many were excluded as cloud-covered, how many of the remainder were predicted damaged and what percentage that represents, the accuracy metrics at the chosen threshold, a precision-and-recall curve across all thresholds, and the estimated total of damaged buildings with its 95 per cent confidence interval. Where no validation labels exist, the report still shows the prediction counts but withholds the accuracy metrics and the estimate rather than presenting an unvalidated number.

Downloads. Results can be exported as a GeoPackage, an open standard file that opens in common mapping software, along with the training artefacts and the building outlines used. This matters for accountability: a partner receiving a HASTE layer can inspect it independently rather than taking a summary figure on trust.

07

DATA SOURCES

Imagery. HASTE holds no imagery of its own; the analyst supplies it. Only GeoTIFF files are accepted. The documentation records that in past activations imagery has come from the following providers.

- Planet, a commercial operator of a large constellation of small satellites that publishes disaster imagery openly.
- Maxar, now Vantor, a commercial provider of high-resolution imagery with its own open data programme for disasters.
- The Airbus Foundation.

- Sentinel-1 and Sentinel-2, the radar and optical satellites of the European Union's Copernicus programme, whose data is free to all.
- Products from the Copernicus Emergency Management Service.
- Aerial imagery from the United States National Oceanic and Atmospheric Administration.

The software itself has no live connection to any of these providers. Placeholder functions for fetching imagery directly from Maxar and Planet exist in the code but are empty. In practice the analyst supplies a link to a file, and for security reasons those links may point only at Azure Blob Storage or Amazon S3, the two hosting services on the platform's permitted list. Public disaster imagery from the major providers is generally published on one of those, which is why the restriction is workable.

HASTE does adjust for the provider in one respect. Different satellites record their colour bands in different orders, and some record bands the human eye cannot see, so the platform holds a lookup table of band orders for Planet Scope, Planet Skysat, Maxar, Sentinel-2, and one partner-specific format, and uses it to assemble a correct colour picture. Where the source is unknown, HASTE falls back to the labels embedded in the file, and failing that assumes the first three bands are red, green, and blue.

Building outlines. Overture Maps by default, described in section 04, and analyst-supplied outlines where better local data exists. HASTE reads the Overture data anonymously from a public store, taking the most recent release available and falling back to a fixed February 2026 release if it cannot determine one. Only polygons are kept.

What HASTE does not use. The platform draws on imagery and building outlines and nothing else. It does not read ground reports, weather data, sensor networks, social media, or population figures. Its picture of a disaster is strictly what a camera in orbit could see, which is a narrower thing than what happened.

Personal data. The platform is not designed to identify people, does not ingest or output personal data, and does not treat person-scale features in imagery as signal.

EVIDENCE OF PERFORMANCE

Everything in this section is the developer's own account. The benchmark results, the deployment record, and the field precision and recall figures all originate from the Microsoft paper and repository. None of them has been independently reproduced or externally validated, here or elsewhere, so they should be read as what the team that built the platform reports about it rather than as third-party verification. That is not an accusation of overstatement. It is the provenance of the evidence, and it is the first thing a sceptical reader is entitled to know.

Validated accuracy. The research paper published alongside the platform, HASTE: A Platform for Rapid Post-Disaster Building Damage Assessment (arXiv:2607.11838), reports experiments on xBD, a public benchmark dataset of paired pre-event and post-event satellite imagery with expert damage annotations. The team collapsed the benchmark's minor, major, and destroyed categories into a single damaged category and measured how well each embedding method performed as the number of labels was varied.

The discrimination score in Table 1 is the area under the receiver operating characteristic curve, usually shortened to AUROC. In plain terms it is the probability that the model ranks a randomly chosen damaged building above a randomly chosen intact one. It runs from 0 to 1, a coin flip scores 0.5, and 1.0 would mean the model never gets a pair the wrong way round. The measure says nothing about where the threshold should sit: a model can rank buildings well and still mislabel a great many of them once a cutoff is applied.

APPROACH	LABELS USED	DISCRIMINATION SCORE
Strongest embedding	1 per cent	0.84
Strongest embedding	10 per cent	0.91
Fully supervised ResNet-50	All labels	0.88

Table 1. Reported performance on the xBD benchmark as the number of labels was varied.

The practical claim is that a handful of labels plus a good general-purpose image description gets close to a conventionally trained model. The headline figure is a modest deficit rather than a match: at one per cent of labels

the score is 0.84 against the fully supervised 0.88, and it is at ten per cent, where it reaches 0.91, that the fast route actually overtakes the baseline. What one per cent buys is not equal accuracy but most of the accuracy, hours sooner and without a machine-learning engineer, which is a real trade and a different claim.

The table does not say which of the two embedding methods produced these scores, and this report cannot resolve it from the published material. Since the lightweight option is the default an analyst would run, and the heavier one is the more capable, that gap matters to anyone reading the figures as a prediction of what they will get.

Deployment record. The figures that follow are evidence of adoption rather than of correctness, and only the Rolling Fork assessment carries an accuracy measurement against field ground truth.

The same paper reports thirty-one field deployments since early 2023, including four cities assessed within three days of the February 2023 Türkiye earthquakes, a tornado assessment in Rolling Fork delivered in under two hours at 0.86 precision and 0.80 recall against field ground truth, and the August 2023 Maui wildfire, where imagery available at nine in the morning yielded an assessment by one in the afternoon identifying roughly 1,700 damaged buildings.

For Hurricane Melissa in Jamaica in late 2025, four areas covering about 2,300 square kilometres were assessed. In Black River, some 110,000 building outlines were examined, of which around 65,000 were obscured by cloud.

Area	Recall	Precision	Estimated damaged
Black River	96 per cent	82 per cent	31,000 buildings
Montego Bay	86 per cent	71 per cent	Not reported

Table 2. Two of the areas assessed during the Hurricane Melissa response.

The variation between Black River and Montego Bay, on the same event with the same team days apart, is substantial, and the very large share of cloud-obscured buildings in Black River is a reminder that a headline damage estimate can rest on a minority of the buildings actually present.

HUMAN OVERSIGHT AND GOVERNANCE

There is no autonomous mode. A person selects the imagery, provides every label the model learns from, reviews the predictions, validates a sample, and decides whether to distribute the result.

Outputs distributed publicly, including through the Humanitarian Data Exchange and partner mapping systems, carry notices warning against over-reliance and encouraging cross-validation against ground reports and other imagery. The notices frame the output as exploratory rather than definitive, and name HASTE, the imagery provider, and the building-outline dataset so that a downstream user can assess provenance for themselves.

The documentation states that where a widespread inaccuracy or pattern of misuse is identified, the laboratory may publish guidance recommending temporary suspension or restricted use of the workflow.

Labels are not reused. Each event's labels belong to that event and are not accumulated into a global training set, which follows directly from the train-per-event design and also limits the accumulation of one analyst's interpretive habits across many responses.

10

LIMITATIONS AND CAVEATS

The repository documents its own weaknesses at length. The most consequential are these.

Every performance figure in this report is developer-supplied. This one is not in the repository's own list, and it is the limitation a sceptical reader should weigh first. As section 08 sets out, none of the reported results has been independently reproduced. The rest of this report reads the source code and reports what it finds, which is first-hand; the performance evidence does not have that standing, and the two should not be given the same weight.

The output depends heavily on who did the labelling. Two competent analysts working the same event from the same imagery can produce materially different results. The documentation gives a concrete example from the Hurricane Melissa response, where an initial set of 153 labels produced predictions that described buildings as around 20 per cent damaged when they

were in fact totally destroyed; the error was caught by visual inspection and corrected by adding a further 107 labels and retraining.

Imagery quality governs everything. Cloud, haze, low light, and imagery captured at an angle rather than from directly overhead all degrade performance, sometimes severely. Planet imagery in particular sits below the roughly 30 centimetre resolution that building-detection models generally prefer, which is part of why the platform leans on external building outlines rather than trying to find buildings itself.

Building-outline coverage is uneven in a way that matters. OpenStreetMap and Microsoft Building Footprints are weakest in the Global South, in informal settlements, in conflict-affected areas, and where construction has been rapid and recent. These are precisely the populations most exposed to disaster. A building with no outline is invisible to HASTE regardless of how badly it was damaged, and the 50 square metre minimum area removes further small structures from the headline figure.

The model does not generalise, by design. A model fitted to one event is not expected to transfer to another, which also means HASTE cannot be operated as a standing monitoring system.

False positives and false negatives have distinct causes. False positives arise from shadows, ordinary construction and demolition unrelated to the disaster, atmospheric artefacts, and vegetation change. False negatives arise from subtle structural damage visible only from an angle, damage hidden by cloud or shadow, and damage at a scale finer than the imagery can resolve.

No contextual data is incorporated. There is no ground truth, no weather, no sensors, no population data. Flood extent is intersected with buildings but water depth is never estimated, so HASTE can say a building is in water and not how deep.

The confidence interval understates real uncertainty. It quantifies the error introduced by sampling and nothing else, as section 05 explains.

Some findings from reading the source code should be added to the project's own list, since they bear on how much weight an output can carry.

There is no genuinely held-out validation data in the trained-model route. The measure used to decide which version of the model to keep is computed on random cut-outs of the same imagery the model was trained on. It is

therefore a measure of fit rather than of generalisation. The independent check in HASTE comes later, from the human validation sample, which is the number a reader should rely on.

The meaning of each class is fixed by its position in a list. Damaged Building is understood downstream as the third class and Cloud as the fourth. An analyst who reorders the classes when setting up a project will get damage figures computed from the wrong category, without any error being raised.

Several defaults disagree between the worked example, the web form, and the server. This is most visible for the number of training passes. Two analysts following the documentation by different routes may not be running the same configuration.

Predictions are matched to buildings by position in a file rather than by an identifier. This is fast, and it means that anything which reorders or filters the building list between steps would silently misattribute damage. The developers are evidently aware of the risk, since the code goes to some trouble to preserve row positions even for buildings it cannot assess.

The platform is not validated for production or autonomous use. It has not been designed, tested, or validated for production deployment or autonomous decision-making, and its outputs are not authoritative damage assessments. The documentation states plainly that users should not rely solely on HASTE outputs for decisions affecting safety, property, or human life, and should not use them to trigger public alerts or resource deployments without independent verification.

11

PRACTICAL NATURE OF THE PLATFORM

HASTE is a full web application rather than a single page or a script. It consists of a browser interface, a set of programming interfaces that the browser front end calls, background workers that handle long jobs such as preparing imagery and training models, a tile server that streams large satellite images into the map view, and a shared Python library holding the analysis logic.

It can be run in two ways. A complete local instance can be started on one machine using Docker, a tool that packages software with everything it needs

to run, which requires no cloud account and is intended for evaluation. A production instance is deployed to Microsoft Azure with a single command, and the deploying organisation controls it entirely.

The local configuration is explicitly flagged as unsuitable for production, since it disables authentication and uses an in-memory storage emulator. A separate hardening checklist is provided for real deployments.

The repository shows active and careful security practice, including automated code scanning and secret scanning, and documented handling of known vulnerabilities in the underlying geospatial libraries where a patched version was not available. Sample data from Hurricane Melissa and the Lahaina wildfire is published so that the platform can be tried without sourcing imagery first.

INTENDED AUDIENCE AND USE

HASTE is aimed at trained humanitarian and disaster-response practitioners working with post-event imagery, and at researchers studying rapid damage-assessment methods. The documentation names non-governmental organisations, United Nations agencies, and government users as the intended downstream consumers of its outputs.

Its stated purpose is to contribute information to preliminary damage assessment in the first hours and days, supplementing rather than replacing expert assessment, and to indicate where damage may be concentrated so that attention can be directed.

It is explicitly not intended for use as any of the following.

- An authoritative damage register.
- Ground truth for insurance, governmental, or public-reporting purposes.
- The sole basis for search-and-rescue tasking or resource allocation.
- An automated alerting system of any kind.

APPLICATION: THE MARIUPOL CORRIDOR SEVERITY MODEL

This section describes work by the Ethical Tech CoLab rather than by the platform's developers, and it describes work in progress. Nothing below has been published as a result, and no HASTE-derived figure currently enters the model it discusses. It is included because the reasoning about when this platform is and is not the right instrument is more useful stated against a real case than in the abstract.

The gap in our own model. The CoLab's Mariupol Corridor Severity Model is a daily civilian-danger index for the 2022 siege, built from six components. The sixth is infrastructure damage, the cumulative share of the city's structures visibly damaged or destroyed, and it is drawn from satellite damage assessments published by UNOSAT, the United Nations Satellite Centre. UNOSAT publishes on particular dates rather than continuously. Across a siege of seventy-seven days the model has five anchor points: 0.02 on 5 March, 0.04 on 14 March, 0.14 on 26 March, 0.32 on 7 May, and 0.33 on 20 May. Every day in between is read off a straight line drawn between them.

The model is candid about what this costs it. In its own words, the word daily describes the resolution of the output and not the resolution of the evidence, and it cannot distinguish 18 March from 19 March on any evidence about what happened on those two days. The practical danger is specific: a stretch of the siege that UNOSAT did not image is rendered as a gently rising line, so an absence of published assessment can be mistaken for an absence of destruction.

Why HASTE is the plausible instrument. The constraint that limits UNOSAT is not imagery, it is the analytic labour of a formal before-and-after building-by-building assessment. HASTE is built for exactly that constraint. It needs only post-event imagery, it fits a disposable model to one event from a small number of hand-marked examples, and it returns a per-building damage fraction with a stated margin of error. In principle it could produce assessments for dates that lie between the UNOSAT anchors, replacing an interpolated line with measured points and letting the damage component move on the evidence rather than on the arithmetic.

What has actually been done. So far, scoping. The model still runs on the five UNOSAT anchors and the straight lines between them, and the per-building display in the tool remains illustrative until real geodata is supplied. We are working through whether HASTE-style assessment of

additional 2022 imagery dates is feasible and, more importantly, whether the result would be honest enough to publish.

What would have to be true first. The obstacles are the ones this report has already described, met in a hard case. They are set out here rather than discovered later.

- Archival imagery. HASTE assesses whatever imagery it is given, and it holds none. A retrospective study needs cloud-free 2022 coverage of Mariupol on the specific dates that would fill the gaps, which is a sourcing problem before it is a modelling one.
- Building outlines. The platform can only attach a prediction to a building it has an outline for, and section 10 records that outline coverage is weakest in exactly the places under greatest pressure. Outlines derived before the siege will also disagree with imagery of a city whose ground and structures have moved.
- The event type. HASTE is benchmarked on xBD, a dataset of disaster damage. Shelling and airstrike damage is not what the reported figures were measured on, and the platform's own design assumption is that a model fitted to one kind of event is not expected to transfer to another.
- Human validation. Every HASTE result depends on an analyst marking training examples and judging an independent validation sample. For a retrospective conflict case that means someone competent to read 2022 imagery of this city, and their judgment, not the model's, sets the ceiling on what the number is worth.
- Status of the output. HASTE describes its outputs as preliminary signals requiring expert validation rather than authoritative damage assessments. The Mariupol model already declares its damage component a lower bound. A HASTE-derived series would have to carry both caveats, not shed them by being newer.

The risk the pairing creates. A denser series looks better evidenced whether or not it is. Replacing five anchors with, say, twenty assessed dates would produce a damage curve that appears to respond to events day by day, and a reader would reasonably infer that the model now knows more about 18 March than it did. It would know more only if each new point were validated to the standard the five UNOSAT anchors carry. Both projects state their uncertainty explicitly and neither should be allowed to launder the other's: HASTE's confidence interval accounts for sampling error and nothing else, and the severity model's daily resolution is a property of its output rather than of its evidence. If HASTE assessments do enter the model, they will be

labelled as such, dated, and reported alongside the UNOSAT figures rather than blended into them.

14

CONCLUSION

HASTE's most useful contribution is not a modelling advance but a reallocation of labour. It moves the machine-learning work out of the way so that the scarce resource, the judgment of someone who can look at an image and know what damage looks like in that country, is applied where it counts: to choosing the imagery, marking the examples, and checking the answer. The model is small, disposable, and fitted to one event, and the platform treats it as such.

The reporting design is equally deliberate. By requiring an independent human validation sample before it will publish an accuracy figure, by separating precision from recall rather than averaging them away, and by attaching a confidence interval to its headline number, the platform makes it harder to mistake a fast estimate for a survey.

The honest reading is that HASTE is fast, transparent about its assumptions, and constrained by things it does not control. Its ceiling is set by the resolution of the imagery available and the completeness of the building outlines beneath it, and both of those are weakest in the places where humanitarian need is greatest. That is not a flaw in the software so much as a statement about the wider data landscape, but it means the platform's value in any given response depends on conditions decided long before the disaster occurred. Read alongside its own documented limitations, it is a credible example of a machine-learning tool built to assist expert judgment rather than to displace it.

References

SOURCES

01 Microsoft AI for Good Lab. HASTE: A Platform for Rapid Post-Disaster Building Damage Assessment. arXiv:2607.11838.

02 Microsoft AI for Good Lab. HASTE, open-source platform released under the MIT Licence.

03 Overture Maps Foundation. Buildings theme, combining OpenStreetMap with machine-derived building datasets from Microsoft and Google.

OpenStreetMap contributors. OpenStreetMap building data.

Microsoft. Global ML Building Footprints.

xBD. Public benchmark dataset of paired pre-event and post-event satellite imagery with expert damage annotations.

MOSAIKS. A generalizable and accessible approach to machine learning with global satellite imagery. Nature Communications, 2021.

Meta AI. DINOv2, a family of vision models trained on 142 million unlabelled images, 2023.

European Commission. Copernicus Emergency Management Service, Rapid Mapping.

European Union. Copernicus Sentinel-1 and Sentinel-2 radar and optical satellite data.

Planet. Open disaster imagery from a constellation of small satellites.

Maxar, now Vantor. Open Data Program for disaster response imagery.

United States National Oceanic and Atmospheric Administration. Emergency response aerial imagery.

OCHA. Humanitarian Data Exchange, the channel through which HASTE outputs have been released openly.

Ethical Tech CoLab. Mariupol Corridor Severity Model, a daily civilian-danger index for the 2022 siege, discussed in section 13.

UNOSAT, United Nations Satellite Centre. Ukraine damage assessments, activation CE20220223UKR, the source of the five anchor points in section 13.