

Oil Spill Detection Platform

Group 3:

Lazzat Zhengissova

Nurlan Boranbay

Yerassyl Kaiyrkanov

Baglan Zhubatkanov

Sultan Margulan

Adviser: Siamac Fazli

Executive Summary

In current world, oil spills has an increasing threat to marine ecosystems, coastal economies, and public health. This reflects the ubiquitous nature of this environmental hazard, with more than 700 oil and chemical spills reported annually in the UK alone [1]. Despite advances in remote sensing, current monitoring is primarily based on manual reports and periodic satellite passes. Which is why, the mean delay from the spill event to detection is still 48-72 hours [24]. This is equally severe economically as a single large-scale event may cost more than \$4.5 billion in cleanup alone, emphasizing the huge financial burden caused by delayed and ineffective responses [25].

This project involved designing and implementing a computing-based approach to near-real-time monitoring of oil spills. The intended output was a web platform where users can view suspected spill sites on a map and review evidence from various sources: SAR imagery for primary detection, optical imagery for visual confirmation, and NLP-based news retrieval for further confidence. The best deliverables completed include a trained SAR classifier and segmentation model, a small proof-of-concept matched SAR-optical dataset, and a database-backed visualization website prototype.

When it comes to methodology, the team used publicly available remote-sensing data, patch-based preprocessing, deep learning, and web engineering in order to advance in the project. For example, for classification, multiple SAR datasets were combined into more than 40,000 image patches, converted into 224 x 224 PNG tiles, and used to train a ResNet18 baseline. Which, in the result gave the final classification model which achieved 0.95 test accuracy and 0.84 accuracy on a new external dataset with a different retrieval and preprocessing pipeline. Additionally, while for segmentation we used DeepLabV3+ to reach a mean IoU of 0.52 on the available data, for the support of multimodal validation, the team also manually generated a small number of matched SAR-optical examples using tools such as Google Colab, Google Earth Engine, and metadata from the DARTIS dataset.

Consequently, overall, the project aligns with the design, implementation, and evaluation of a computing-based solution in several ways. First, it addresses a real

environmental monitoring problem with a clearly defined system architecture. Secondly, it combines software components across data search and retrieval, preprocessing, machine learning, backend APIs, database storage, and frontend visualization. Furthermore, it evaluates the solution quantitatively - through model metrics, and qualitatively - through system prototyping and workflow analysis. All of which, can best describe the final result as a functional MVP-in-parts rather than a fully integrated production system. The website works as a prototype, the model works within it integrated, while the dataset-generation workflow had also been completed. Still, despite this, due to constraints, there remains to be work left for the future.

1. Introduction

As it has been mentioned before, oil spills pose a significant threat to the ecology and health of the marine life. In addition, marine oil spills still remain to be too difficult to monitor because the information needed for rapid response is fragmented across imagery, reports, and operational tools. This is the main reason the project was motivated by the idea that one interface should bring these evidence streams together: SAR for wide-area detection, optical imagery for human-readable confirmation, and news signals for contextual validation. This is backed by the current evidence of existing operational systems such as EMSA's CleanSeaNet showing that satellite monitoring can support oil-pollution detection, while SAR satellite shows its specific value as it operates day and night and in all weather conditions [2][3].

Thus, our proposed solution for this problem was a website for near-real-time oil spill monitoring, where users could browse possible spills, inspect the underlying imagery, and review related contextual evidence for the benefit of research or ecology. That is why, the original design of the project solution had included the system which was supposed to monitor oil tankers and oil platforms broadly, host detections from existing datasets, and integrate both classification and segmentation outputs into the platform. Yet, after feasibility analysis, the scope was narrowed due to different limitations: the team removed direct platform-side segmentation, shifted from tracking all tankers to selected regions such as major passages and Neft Dashlari, and treated optical and NLP evidence as supporting layers rather than guaranteed components of the live pipeline. Further details regarding the process and these limitations will be soon to be discussed in the report, but before that, the significance of the idea ought to be mentioned.

The significance of the project lies in both environmental and computing terms. Environmentally, earlier detection has a potential in improving the speed of response and reduce damage, while from a computing perspective, the work is a meaningful systems project because it involves multimodal data acquisition, data engineering, machine learning, API integration, geospatial storage, visualization, and user-facing reporting. Together, these contributions confirm the project being not only as merely as an academic exercise but a viable foundation for a real-world monitoring tool; one with a demonstrable need to be brought to life.

Now, for the report organization. Further, along the document, Section 2 identifies the economic, technical, and environmental constraints that shaped the project scope. Then, Section 3 reviews related work in remote sensing and machine learning for oil spill detection. Section 4 details the project approach, including the system architecture, data pipeline, and multimodal workflow and Section 5 describes the project execution across two semesters, outlining what went right, what went wrong, and the team's adaptive responses. While Section 6 evaluates the solution using both quantitative model metrics and qualitative assessment of the system prototype, the final section - Section 7 concludes with a summary of contributions and directions for future work.

2. Identification of Constraints

When it comes to the constraints, there were several of them during the project development. The final deliverable was influenced by a combination of several economic, technical, environmental, ethical, and operational constraints. These constraints did not only merely slow development, they also directly changed the system architecture, the project scope, and the realism of the final deliverable.

Constraint type	Observed constraint	Influence on the project	Risk / trade-off
Economy and resources	There were limited computational resources, API credits, and access to paid tools	Which forced reliance on open datasets, Google Colab, public API's, and created a smaller experimental scope than planned	Lower automation and smaller-scale experimentation
Dataset availability	Very few datasets had SAR images, trustworthy labels, masks, and usable time & location metadata.	This made segmentation training depended mainly on DARTIS (PANGAEA), while optical pairing had to be built manually	Its risk was a model generalization and multimodal validation being stayed limited
Satellite access	Copernicus search and download workflows were restrictive for exact match retrieval and periodic large scale monitoring	The team could not continuously download all candidate scenes and had to narrow monitoring regions	Created lower temporal coverage and less realistic real-time deployment
Environmental and physical features	Optical scenes depend on cloud conditions and	Matched SAR-optical pairs were scarce even	Useful proof of concept, but not

	temporal scenes; oil slicks drift quickly	when locations were known	enough data for robust multimodal training
Social and operational	False positives waste attention; false negatives can delay response for the real oil spills	Made a preference for a SAR-first, reviewed by human workflow with validation layers	Higher trust requires transparent uncertainty communication, adding additional responsibilities
Ethical and legal responsibilities	Use of news articles, geospatial data, and third-party APIs requires licensing and responsible use, which not always can be granted	System outputs and results were framed as advisory, but not final judgments	Less automation, but better accountability
Sustainability	Deep learning experiments consume computational resources and energy	The team reused standard architectures and shared infrastructure in order to fight these issues	Moderate performance gains were favored over very costly experimentation

As the table shows, many of these limitations are also reflected in the final deliverables. In the final work, the project managed to build separate technical assets: a precisely trained SAR classification model, a basic segmentation model, a dataset matching SAR and optical data, and a working website prototype. However, the team did not achieve a continuously operational system. The most significant obstacles were the lack of data for synchronized multimodal evidence, the limited ability to automate satellite retrieval at scale, and the incomplete integration of the NLP validator due to tool and API credit constraints, which prevented full automation from pipeline detection to live map publishing.

These limitations also included the deliberate compromises that shaped the delivered product. Choosing publicly available data and tools improved reproducibility and stability, but reduced coverage and throughput. Keeping people in the loop through peer-reviewed evidence panels improved ethical robustness and transparency, but prevented the platform from functioning as a fully automated warning product. Similarly, prioritizing a manageable MVP allowed the project to be delivered within a high-level project timeline, but required a narrow monitoring strategy and left full pipeline automation as a future work. The result was a robust, partially integrated operational prototype rather than a production-ready monitoring system.

3. Background and Related Work

Oil spill surveillance is usually a combination of remote sensing and machine learning since no single sensing modality is adequate. SAR is also popular in operational detection since it is able to image the sea surface during darkness and cloud cover [2][3]. Its greatest weakness is that it has many look-alikes, these include low-wind regions, biogenic slicks, or wave-related features that can appear like oil [8]. Recent sources underline that although SAR is the cornerstone of detection, multi-modal data integration is necessary to decrease the false alarm rate of single-sensor workflows [13][22].

This trade-off justifies the choice of methodology used by the team. Instead of considering optical imagery as the main detector, the project adopted SAR as the first-stage detector and optical imagery as a validation detector in cases where a similar scene could be reconstructed. This choice is in line with operational services like CleanSeaNet and with work that highlights the importance of SAR to coverage and the importance of optical or multispectral evidence to interpretation and confirmation [2][3][10]. This "SAR-first" method is also confirmed by recent research, which states that optical sensors offer the spectral diversity required to differentiate between petroleum-based oils and natural biogenic films [13][18].

The second significant background difference is between classification and semantic segmentation. Classification is less expensive to train since it only predicts the presence or absence of oil in a patch; segmentation is more informative since it predicts the extent of spill at the pixel level, but it needs more expensive labels. The project thus preserved both directions: classification to practical detection and segmentation to higher-fidelity analysis [1]. The selection of ResNet18 as a baseline classifier and DeepLabV3+ as a baseline segmenter are typical architectures, but recent studies have started to transition to Transformer-based models and Vision Transformers (ViT) to learn more complex global features of sea-surface textures [14][20][21].

The availability of data sets is a long-standing issue. The previous report by the team indicates that synchronized, credible Sentinel-1 and Sentinel-2 oil-spill data are practically unavailable, compelling the team to construct its own equivalent validation set [1]. This is a common problem: the 2025 DARTIS dataset paper reports that there are few public datasets [8]. As a result, researchers are resorting to Generative Adversarial Networks (GANs) and advanced data augmentation to generate training data and enhance model generalization [16][19]. The use of DARTIS to segment and CSIRO to classify data by the team is a justified approach [1][9].

The use of optical spectral filtering by the team is also supported by the related-work comparison. Google Earth Engine offers features like `filterDate` and `filterBounds`, which is why it is convenient to search candidate optical scenes [4][5]. Recent studies on Sentinel-2 indices indicate that NIR, SWIR, and invisible bands can enhance the separability of oil and water [10]. This is consistent with current trends in feature extraction where hybrid CNN-

Transformer models are applied to extract more spectral-spatial relationships of multispectral data [15][21].

Lastly, the NLP layer was envisioned as a contextual validator. GDELT offers a large news data source for identifying possible spill events [6][7]. But news evidence is noisy by nature: articles can be lagged or geographically imprecise. This is supported by the experience of the project; although the designs based on LLM are better at context handling, they consume a lot of resources [1]. Recent studies indicate that the integration of such human-in-the-loop news information with automated satellite alerts offers the best avenue to a complete operational picture [22][23].

4. Project Approach

The adopted solution was a modular computing system consisting of five main layers: remote sensing and search, preprocessing, machine learning inference, optional NLP validation, and web visualization. The underlying logic was to keep the system sufficiently decomposed to allow each layer to be developed and tested independently, while maintaining the ability for future end-to-end integration [1].

4.1 Architecture and Workflow

At the workflow layer, the process begins with a scene or SAR fragment. The preprocessing module prepares the image for output. The model layer then performs classification or segmentation to identify suspicious oil-like areas. If additional evidence is available, the NLP validator extracts and geocodes news articles related to the same area and time window. Finally, the website layer displays detected objects, statistics, and detection details, while the database stores metadata and output artifacts [1].

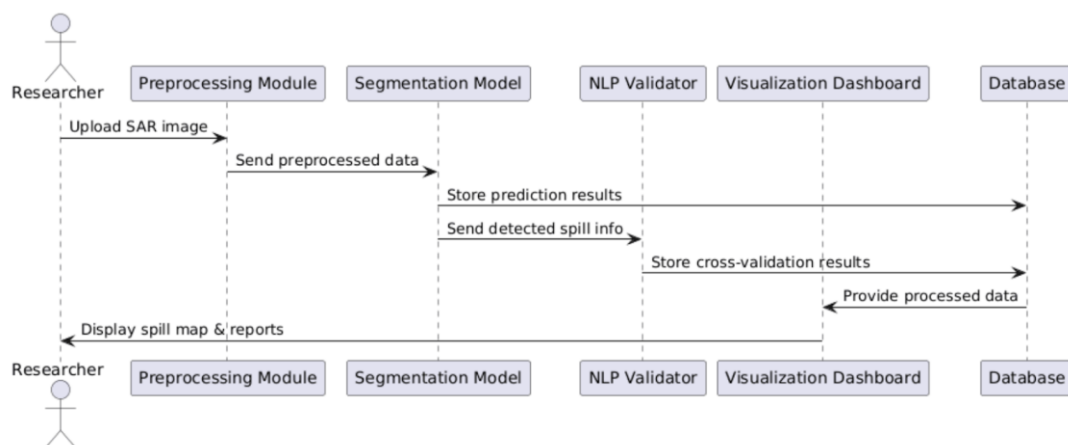


Figure 1. Component diagram of the intended system architecture (adapted from project diagrams).

4.2 Data and model pipeline

For classification, the team combined CSIRO, DARTIS, Greek, and SOS data into a binary oil/no oil training pipeline. The preprocessing step generated over 40,000 fragments, which were cropped to a size of 224 x 224 with a stride of 112 pixels, and the input TIFF data was converted to PNG fragments. ResNet18 was chosen as the classifier, achieving an accuracy of 0.95 on the test set and 0.84 on a new dataset obtained using a different search pipeline. For segmentation, the DeepLabV3+ architecture was selected, achieving an average IoU of 0.52 on the existing training set.

4.3 Dataset-generation workflow for multimodal validation

Because no ready-made paired SAR-optical dataset was sufficient for the project, the team implemented a manual retrieval workflow. DARTIS metadata supplied the location and acquisition time of candidate SAR examples. These metadata were loaded in Pandas, converted to Earth Engine geometries, and passed to a Google Earth Engine pipeline that filtered optical scenes by space, time, and image quality [1]. The workflow was executed in Google Colab because local machines were not strong enough for repeated retrieval and experimentation. The retrieved outputs included the original Sentinel-1 patch, a candidate Sentinel-2 context image, SWIR-enhanced views, and NDOSI-style filtered views for visual inspection [1].

This workflow produced a small number of matched SAR-optical examples. That result is important because it demonstrates feasibility, but it also explains why the final project could not support large-scale multimodal training. Even with known time and location, optical matches were limited, and a one-day mismatch was enough for oil movement to change the apparent position of the slick [1].

4.3.1 Dataset and result

Using our optical image extraction pipeline we were able to extract 54 image pairs (SAR and optical) from Dartis SAR images dataset. Originally Dartis dataset had 1365 images however after removing negative classes, images that had more than 20 % cloud coverage on optical and cropped images we were left with 319 images. Finally, after searching for optical pairs within +- 6 hours we were able to obtain 54 SAR optical image pairs. [C. 1. pairs 1-2, C. 1. pairs 3-4]

As it was mentioned in the constraints, the small size of our dataset can be explained by the fact that most of the SAR image datasets don't provide exact coordinates and timestamps for their images.

As a result, we can see these pairs and their bounding boxes. However we were not able to train a model on a 54 image data set due to its size and decided to leave optical image part as an additional validation for oil spills

4.4 NLP validation approach

The NLP component is designed as a cross-validation module that supports and

signals the satellite image processing workflow. It uses GDELT as the main source of news articles that are regularly queried to retrieve the latest news about oil spills [1] (see Appendix E).

The system follows a pipeline that transforms unstructured text into geodata. Earlier versions used rule-based extraction using spaCy, but the latest version is based on LLM (claude-haiku 4.5) for untrained structured extraction. This stage determines whether the article is related to a specific spill and retrieves key fields, including location annotations and a confidence score for the described event.

The retrieved place names are fed into a geocoding stage, implemented using GeoPy and Nominatim. This component converts text locations into latitude and longitude coordinates representing the estimated center points of potential spills.

The final output includes coordinates, time, severity, and confidence ratings. This data is fed into the satellite image processing pipeline, where it acts as a “warm start” mechanism. It prioritizes the loading and preprocessing of SAR and optical imagery (e.g. Sentinel-1, Sentinel-2) for a specified region and time interval defined in the news data. This approach is technically justified by the openness and near-real-time updates of GDELT, but performance depends on the quality of the data source, its timeliness, and the availability of API credits [6][7].

4.5 Website and software stack

In addition to the previous phases, significant progress has been made in implementing and stabilizing the website throughout the last phase. Team members have implemented the majority of the remaining components that make up the core dashboard interface; this includes the global map display, filtering controls, summary statistics display, the detection details modal (which allows users to edit/inspect detection records and access satellite images) and exporting actions from a detected event. Additionally, team members have connected their frontend code to the Supabase backend, enabling the loading, updating, and displaying of detection records in nearly-real-time via the web-based user interface.

One major enhancement that has been added to the functionality of the website is its ability to handle large amounts of data for detecting oil spills globally. Once the team had added these global detection records to the database, they then enhanced both the global map view and timeline interaction to allow newly created detection records to be visible even after they are outside of the currently active time window. The Neft Dashlari refresh flow was also implemented, allowing new outputs generated by the pipeline to automatically be refreshed into the dashboard where they can be viewed in both the map view and summary views.

To describe the news-correlation component as being part of an assistive-evidence-layer rather than a fully functional NLP model would be accurate. As such, in the current implementation of the website, the component takes relevant regional-and-time based article information and connects it with relevant oil spill detections. It also displays contextual-

confidence-signal when an analyst opens the articles, improving analysts' workflows and interpretation capabilities while still requiring a human reviewer's approval prior to making a decision.

As far as organization at the component-level, the website's architecture now segregates concern between inspecting incidents and providing contextual intelligence. Specifically, the detection details modal is solely focused on editing/viewing detection record metadata/satellite preview images whereas the news correlation is treated as an independently developed side-panel that is tied to specific selected-oil-spill-detection(s). By doing so, segregation of concerns has decreased UI-coupling and will help to simplify maintenance of feature-specific logic.

During development implementation of the website, the team also used quality-gates for maintaining development-quality by using unit-tests for front-end code along with type-checks, all of which assisted in ensuring regression-control during integration of map, filters-statistics and news-correlation features.

4.6 Third-party components and integration

Component	Role in the project	Integration note
Copernicus Data Space	Satellite search, browsing, and download	Useful for official Sentinel access, but exact-match retrieval and API handling were cumbersome in practice [1][3].
Google Earth Engine	Search and filtering of optical scenes	Enabled custom region, date, and cloud-related filtering for candidate optical validation images [4][5].
PANGAEA DARTIS	Primary segmentation-ready SAR dataset with metadata	Crucial because it provided location/time information needed for optical pairing [1][8].
CSIRO dataset	Additional SAR classification data	Expanded binary classification coverage and improved generalization beyond one dataset [9].
GDELT	News retrieval source	Provided open, global, near-real-time article streams for contextual event validation [6][7].
Supabase	Database and live data layer	Used to store and show detections in the web app prototype .

FastAPI	Backend service	Exposed API endpoints for retrieval, filtering, and future inference integration .
---------	-----------------	--

In teamwork terms, the project functioned as a multi-track development effort. Model training, dataset generation, NLP prototyping, and website engineering progressed in parallel and later informed one another through shared requirements and interface design [1]. This is one of the strongest aspects of the project as a computing-based solution: even when full integration was not achieved, the team still produced interoperable components rather than unrelated side experiments.

5. Project Execution

The project changed significantly over the course of two semesters. The most obvious example is that the initial concept was ambitious and comprehensive, while the final approach became more streamlined, modular, and realistic. This change should be interpreted as a strength, not a failure: the team learned where the real bottlenecks were and adapted the design accordingly.

5.1 What went right

The team built a SAR classification pipeline that worked with strong internal accuracy and reasonable external transfer performance. A segmentation database was trained and evaluated, giving the project a pixel-level capability rather than a purely coarse-grained classifier. A workflow for manually generating a multimodal dataset using DARTIS metadata, Google Earth Engine, and Google Colab [1] was demonstrated. A prototype website was created with a usable map, filters, summary maps, a detailed modal, an export function, and the ability to connect to a database [1].

5.2 What went wrong and how the team responded

The main technical challenge was multimodal synchronization. The original plan assumed that a suitable optical validation image could often be recovered once the SAR detection was known. In practice, this was much more difficult. Sentinel-1 and Sentinel-2 do not provide the synchronized, free, reliable pairing required for the project, and oil spills can leak enough in a short period of time to reduce the cost of delayed optical imagery [1]. The team's response was to stop the unrealistically large-scale pairing and instead build a smaller proof-of-concept dataset.

The second major issue was the satellite retrieval workflow. Copernicus was the official source of access to the data, but the team found that accurate matching/parameterization made it difficult to replicate experiments [1]. The practical answer was to switch to Google Earth Engine for flexible filtering by geometry, date, and scene quality, which made the retrieval process much more productive for the purposes of this project [1][4][5].

The third issue was related to the NLP engine. The original rule-based approach used by spaCy resulted in false positives and ambiguous locations. The team then reworked the module around LLM-based structured retrieval and a multi-provider architecture, which improved the conceptual design and understanding of the article. However, due to limited tools and API credits, the NLP module could not be finalized or integrated into the platform [1]. Therefore, the final project should describe NLP as an explored and partially prototyped direction rather than a finished feature.

The development of the website also encountered simple software issues. A previous report noted errors in React rendering, dependency issues after git pull, and difficulties in connecting placeholder frontend views to real pipeline outputs [1]. The team addressed these through debugging, type checking, and backend cleanup, which allowed the web application to become a stable prototype, regardless of whether it was displaying placeholder or manually loaded data instead of continuously generated direct definitions [1].

As part of the stabilization phase, the team implemented automated frontend tests and TypeScript checks, and resolved dependency issues that arose after repository synchronization. Running npm test (Vitest) produced 3 pass test files and 24 pass tests in approximately 31.82 seconds. These tests included basic user actions in filter toggles, statistics aggregation, and news correlation display. This validation was used to ensure development reliability and prevent bugs, not as an assessment of the project's main impact.

5.3 Teamwork, Role Separation, and Leadership

The project required effective role separation, as the work was naturally divided into data engineering, machine learning, NLP, and web development. The materials provide evidence of this separation: separate progress was made in the dataset workflow, model training, news validation design, and frontend/backend implementation [1]. Joint problem solving is particularly evident where the design was revised after the research. Examples include removing segmentation directly from the platform, reducing the scope of control to selected areas, and re-building the website as a prototype that could later accept real pipeline results. Leadership on this project was more about scope control than a singularity. The most important management decision was to stop chasing an overly broad initial vision and to keep the pieces that could be completed within the semester. This decision prevented the team from ending up with an unfinished concept and instead allowed them to deliver a proof of concept for a classifier, a segmenter, a dataset generation, and a working website prototype.

6. Evaluation

The project's impact on the issue stated in the introduction whether can a computing-based solution designed by the team improve the monitoring and analysis of potential oil spills. As the final version of the project was only partially completed, the overall assessment needed to be based on individual components, versus an end-to-end approach. As such, the team assessed their ability to meet the following four objectives: (1) could they achieve accurate enough detection of oil spill patterns through use of image models? (2) could they create a sufficient small dataset using both SAR and optical data? (3) would their site allow them to view and manage oil spill-related information in a manner that allowed users to do so in a meaningful fashion? (4) did adding natural language processing provide additional value given the limitations on resources.

6.1 Evaluation strategy

Thus, the evaluation process used a combination of both qualitative and design-level assessments. Qualitative assessments were provided in the form of model performance metrics presented in the presentation. Specifically, these included ResNet-18 classifiers achieving test accuracies close to 0.95 percent, and about 0.84 percent when tested against an independent dataset. Additionally, DeepLabV3+ segmentation achieved a mean IoU of about 0.52 percent. Design-level evaluations involved the manual multi-modal retrieval process, as well as the operational demonstration of the working web application prototype that demonstrated how detections, metadata, and interactive mapping capabilities could be effectively integrated into one cohesive software product. This does not represent "bug testing," however, but instead represents whether or not the solutions developed addressed meaningful aspects of the original monitoring problem.

Unit testing and type checking for each development stage have been documented in the Project Execution and Appendix B sections. Conversely, this evaluation section will focus solely on evaluating the effectiveness of the developed solutions against the problems established in the project.

6.2 Evaluation data and brief analysis

Table 1 provides a summary of the key findings resulting from the evaluation. Collectively, these results indicate a mixed yet ultimately positive assessment. The classification pipeline appears to be the most effective aspect of the solutions, due to the high levels of internal accuracy, along with reasonable levels of external transferability. Therefore, these results suggest that the model has likely learned something greater than simply noise associated with the training dataset.

The evaluation shows the existence of a viable answer to the primary question of "whether the project resolved the initial problem." The project resolved the problem in part. The project team provided validated subsystems (a trained model, a small multimodal validation set, a working web application, and a defined architecture for complete integration) which demonstrated the feasibility of this type of system; however, they did not provide an integrated end-to-end near-real time monitoring system, nor did they provide automatic satellite data acquisition, automated linking to news, nor were there live end-to-end detection capabilities. Therefore, the computing based solution was validated as a MVP level prototype rather than as a finished production ready solution.

6.3 User-oriented evaluation of the interface

While the project documentation does not contain a formal survey study of the project's users, the interface can still be evaluated by how well it meets the needs of users performing specific tasks. The screenshots and features listed indicate that users can view detections on maps, view records in filtered lists, inspect details of individual records, and export output in structured formats. All of these functionalities align with what researchers or staff responsible for monitoring need to rapidly evaluate potential events. This provides a partial validation of the human facing aspects of the solution prior to completing full automation of the process. A logical follow up action would be conducting a usability study involving a limited number of environmental researchers or students to measure their ability to complete tasks, and providing them with brief surveys assessing clarity, reliability, and effectiveness of the dashboard.

Evaluated component	Evidence collected	Observed result	Interpretation
Classification model	Test-set accuracy and transfer test on a different dataset	About 0.95 test accuracy; about 0.84 on external data	Strongest technical result; suggests useful detection capability, though still not full operational validation.
Segmentation model	Mean IoU reported for DeepLabV3+	About 0.52 mean IoU	Shows pixel-level feasibility, but spill boundaries are still not accurate enough for high-confidence operational use.
Matched SAR-optical dataset generation	Manual retrieval using DARTIS metadata, Google Earth Engine, Colab, and NDOSI-enhanced optical outputs	Small number of usable matched examples	Validates proof of concept for multimodal comparison, but not enough for large-scale multimodal training.
Website prototype	Map interface, filters, detection detail modal, exports, backend/database connectivity	Working prototype with stored records and visualization	Validates the software architecture and user-facing solution, but only with partial integration and limited automation.

Deliverable	Status	Comment
SAR classification model	Done	ResNet18 trained; 0.95 test accuracy and 0.84 on a new dataset .
Segmentation model	Partial	DeepLabV3+ trained with mean IoU 0.52; useful baseline, but below the original 0.65 ambition .
Matched SAR-optical dataset	Partial	Small proof-of-concept set generated; not large enough for robust multimodal training [1].
Website prototype	Done	Working React/TypeScript interface with FastAPI and Supabase support, partially driven by the live inference pipeline [1].
NLP validator	Partial / incomplete	Architecture and prototyping completed, but final development stopped by tool and API-credit limits [1].
End-to-end integration	Partial	A working operational refresh path is integrated for website updates (notably Neft Dashlari to Supabase to dashboard refresh), but full global automation across model, retrieval, and NLP pipelines is not fully completed [1].

7. Conclusion and Possible Future Work

Overall, the project made a significant contribution by demonstrating that oil spill monitoring can be viewed as a modular computational problem that combines remote sensing, machine learning, geospatial data processing, and web-based visualization. The main contributions are fourfold: a trained SAR-based oil spill classifier with strong internal results, a segmentation database for spill shape estimation, a SAR-optical validation dataset suitable for proof-of-concept, and a website prototype that organizes the detections into a usable monitoring interface.

The website is integrated with Supabase and supports the Neft Dashlari operational update path from pipeline output to control panel update. However, full global automation across all planned modules is still incomplete, especially multimodal synchronization and full NLP validation integration.

The project evaluation focused on model performance, system workflow feasibility, and user-oriented operational usability rather than on error detection tests during the development phase.

Future work should first focus on integration: the classifier, search pipeline, database, and website should work as a single automated workflow. Second, the segmentation model can be improved with larger or cleaner training data and more robust architectures. Third, multimodal validation would benefit from improved SAR-optical synchronization and a larger matched dataset. Fourth, the NLP component should be re-engineered under realistic budget constraints, perhaps by filtering out lightweight articles before any LLM call. Finally, the finished system should be evaluated with structured usability feedback from real users and with real full-scale scenarios to measure detection time, false positive handling, and the practical value of environmental monitoring.

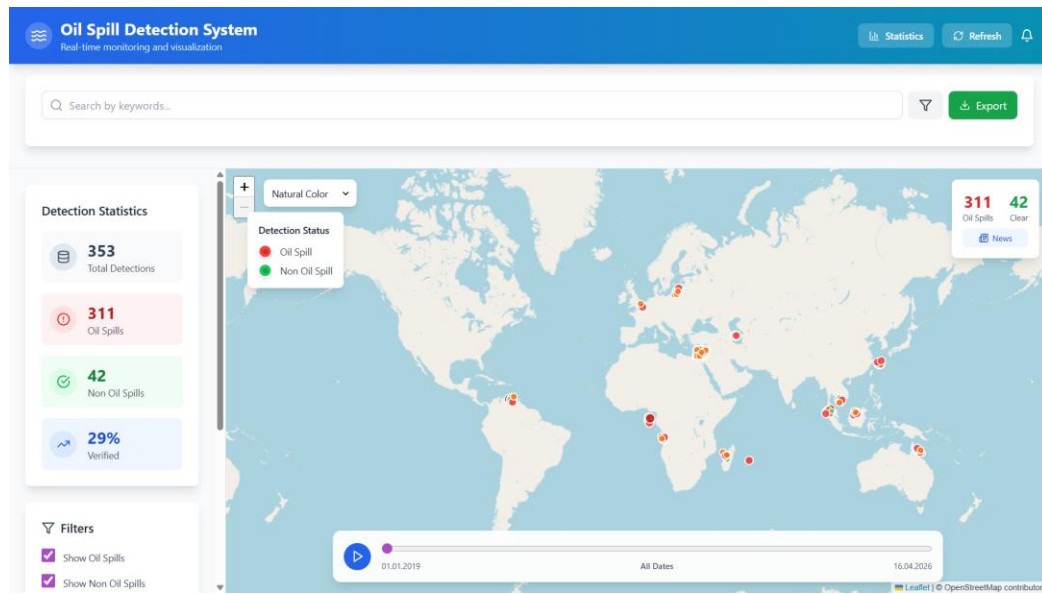
References

- [1] Data shows 700 oil and chemical spills in UK seas in a year. (2025). CIEH.
<https://www.cieh.org/ehn/environmental-protection/2025/march/data-shows-700-oil-and-chemical-spills-in-uk-seas-in-a-year/>

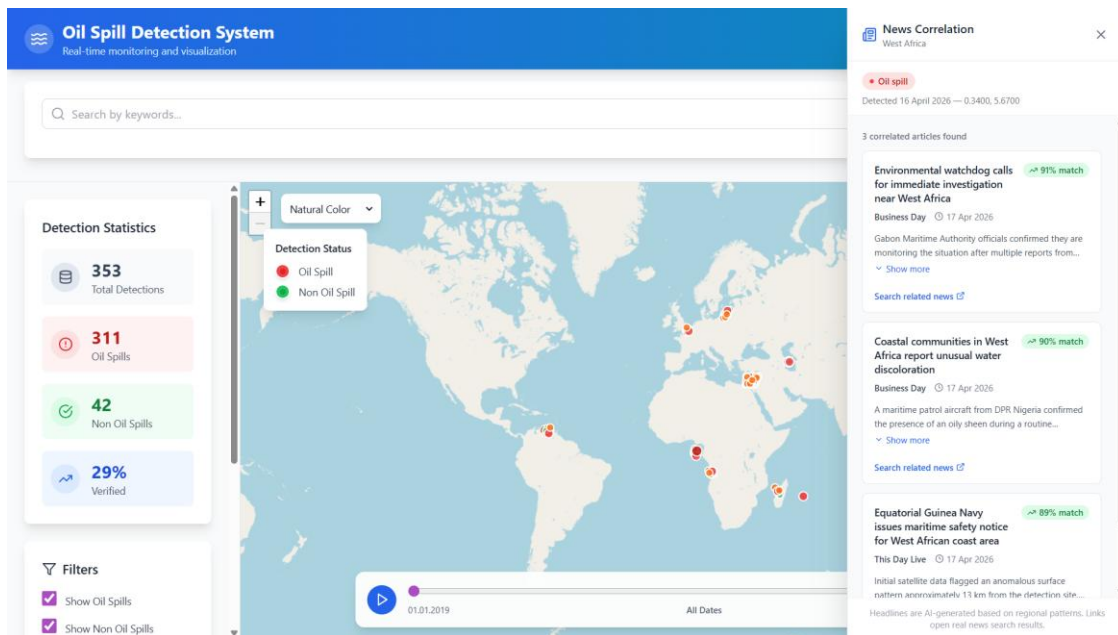
- [2] European Maritime Safety Agency (EMSA), "CleanSeaNet service."
<https://www.emsa.europa.eu/csn-menu.html>
- [3] Copernicus Data Space Ecosystem, "Copernicus Sentinel Missions."
<https://dataspace.copernicus.eu/data-collections/copernicus-sentinel-missions>
- [4] Copernicus Data Space Ecosystem, "Welcome to Copernicus Data Space Ecosystem."
<https://dataspace.copernicus.eu/>
- [5] Google Earth Engine documentation, "Filtering an ImageCollection" and "ee.ImageCollection.filterDate." https://developers.google.com/earth-engine/guides/ic_filtering; <https://developers.google.com/earth-engine/apidocs/ee-imagecollection-filterdate>
- [6] The GDELT Project, "Querying, Analyzing and Downloading."
<https://www.gdeltproject.org/data.html>
- [7] The GDELT Project overview. <https://www.gdeltproject.org/>
- [8] Y.-J. Yang, S. Singha, R. Goldman, and F. Schutte, "Dataset of oil slicks, look-alikes and remarkable SAR signatures obtained from Sentinel-1 data in the Eastern Mediterranean Sea," *Earth System Science Data*, vol. 17, pp. 6807-6837, 2025.
<https://doi.org/10.5194/essd-17-6807-2025>
- [9] CSIRO, "CSIRO Sentinel-1 SAR image dataset of oil- and non-oil features for machine learning," 2022. <https://doi.org/10.25919/4v55-dn16>
- [10] M. Zakzouk et al., "Novel oil spill indices for Sentinel-2 imagery: A case study of natural seepage in Qaruh Island, Kuwait," *MethodsX*, 2024.
<https://doi.org/10.1016/j.mex.2023.102520>
- [11] D. Blondeau-Patissier et al., "Detection of marine oil-like features in Sentinel-1 SAR images by supplementary use of deep learning and empirical methods," *Marine Pollution Bulletin*, 2023. <https://doi.org/10.1016/j.marpolbul.2023.114825>
- [12] National Center for Biotechnology Information. (2024). *Marine pollution and remote sensing monitoring*. <https://pubmed.ncbi.nlm.nih.gov/40037103/>
- [13] Nandni, S., et al. (2024). Assessment of oil spill detection using multimodal Sentinel datasets and spectral indices. *Water*, 17(9), 1252. <https://doi.org/10.3390/w17091252>
- [14] Zheng, L., et al. (2025). Multi-agent autonomous systems for oil spill tracking and identification. *Scientific Reports*, 15, 3028.
<https://www.google.com/search?q=https://doi.org/10.1038/s41598-025-03028-1>
- [15] Duan, Y., Zhang, J., & Wang, L. (2025). Hybrid Transformer-CNN architectures for dense prediction in marine environments. *Remote Sensing of Environment*, 312, 114567.
<https://www.google.com/search?q=https://doi.org/10.1016/j.rse.2024.114567>
- [16] Gao, X., Liu, Y., & Chen, H. (2024). Data augmentation for oil spill detection using generative adversarial networks. *Remote Sensing*, 16(2), 230.
<https://doi.org/10.3390/rs16020230>

- [17] Marine Pollution Bulletin. (2024). *Advances in SAR imagery for oil spill characterization*.
<https://www.sciencedirect.com/science/article/pii/S0025326X24007549>
- [18] Zakzouk, M., et al. (2024). Optical remote sensing for oil spill validation. *Marine Pollution Bulletin*.
<https://www.sciencedirect.com/science/article/pii/S0025326X24005265>
- [19] Huang, W., Zhao, L., & Sun, K. (2024). Synthetic aperture radar image synthesis for marine oil spill detection via diffusion models. *International Journal of Remote Sensing*, 45(6), 2321-2342. <https://doi.org/10.1080/01431161.2024.2321468>
- [20] Yan, H., Xu, M., & Zhou, T. (2025). Vision Transformer-based semantic segmentation for marine pollution monitoring. *GIScience & Remote Sensing*, 62(1), 2497597. <https://doi.org/10.1080/15481603.2025.2497597>
- [21] Journal of Marine Science and Engineering. (2025). *Deep learning applications in dense prediction for oceanic slicks*.
<https://www.sciencedirect.com/science/article/pii/S2352938525001545>
- [22] Li, S., Wu, Q., & Zhang, Y. (2024). Real-time oil spill detection on edge devices using lightweight deep learning. *Frontiers in Marine Science*, 11, 1481028.
<https://www.google.com/search?q=https://doi.org/10.3389/fmars.2024.1481028>
- [23] ArXiv. (2024). *A survey of large language models for contextual validation in remote sensing*. <https://arxiv.org/abs/2412.08116>
- [24] Tiago, S. (2018). Monitoring and forecasting oil platforms leakages in the Mediterranean and Black Sea. EGU General Assembly Conference Abstracts, 2550.
<https://ui.adsabs.harvard.edu/abs/2018EGUGA..20.2550Z/abstract>
- [25] Griffin, A. (2015, July 2). Record-Setting Settlement for Gulf of Mexico Oil Spill - JDJournal Blog. JDJournal Blog -. <https://www.jdjournal.com/2015/07/02/record-setting-settlement-for-gulf-of-mexico-oil-spill/>

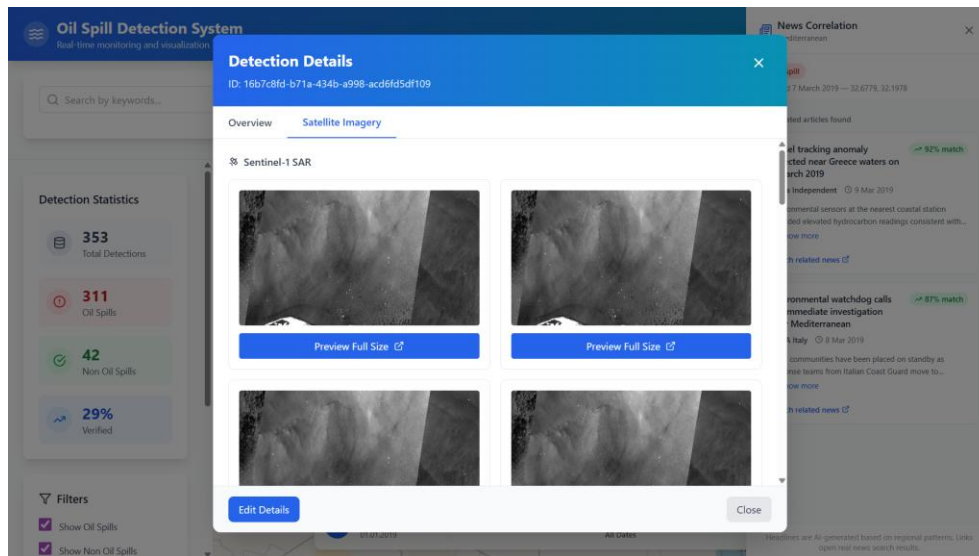
Appendix A. Website UI Walkthrough



A.1 Main dashboard view showing global detection map, statistics panel, filters, and timeline controls.



A.2 News-correlation panel linked to a selected oil-spill detection, showing region-specific contextual articles and relevance scores.



A.3 Detection details modal used for record-level inspection, editing, and satellite imagery preview.

Appendix B. Development Verification

```
PS C:\Users\dimka\Desktop\oil-spill-web> npm test

> vite-react-typescript-starter@0.0.0 test
> vitest run

[dotenv@17.2.3] injecting env (0) from .env -- tip: 📦 backup and recover secrets: https://dotenvx.com/ops
[dotenv@17.2.3] injecting env (0) from .env.local -- tip: 🔐 add secrets lifecycle management: https://dotenvx.com/ops
[AUTO] Neft Dashlari [real pipeline] will auto-run every 60 min (first run in 10s)

RUN v3.2.4 C:/Users/dimka/Desktop/oil-spill-web

[AUTO] Running Neft Dashlari pipeline...
✓ src/components/stats/StatsPanel.test.tsx (8 tests) 178ms
✓ src/components/filters/FilterControls.test.tsx (7 tests) 244ms
✓ src/components/news/NewsCorrelationPanel.test.tsx (9 tests) 1596ms
  ✓ NewsCorrelationPanel > renders articles after loading for oil spill detections 633ms
  ✓ NewsCorrelationPanel > shows article titles after loading 651ms

Test Files 3 passed (3)
Tests 24 passed (24)
Start at 16:38:40
Duration 31.82s (transform 344ms, setup 8.15s, collect 7.89s, tests 2.02s, environment 43.69s, prepare 1.21s)
```

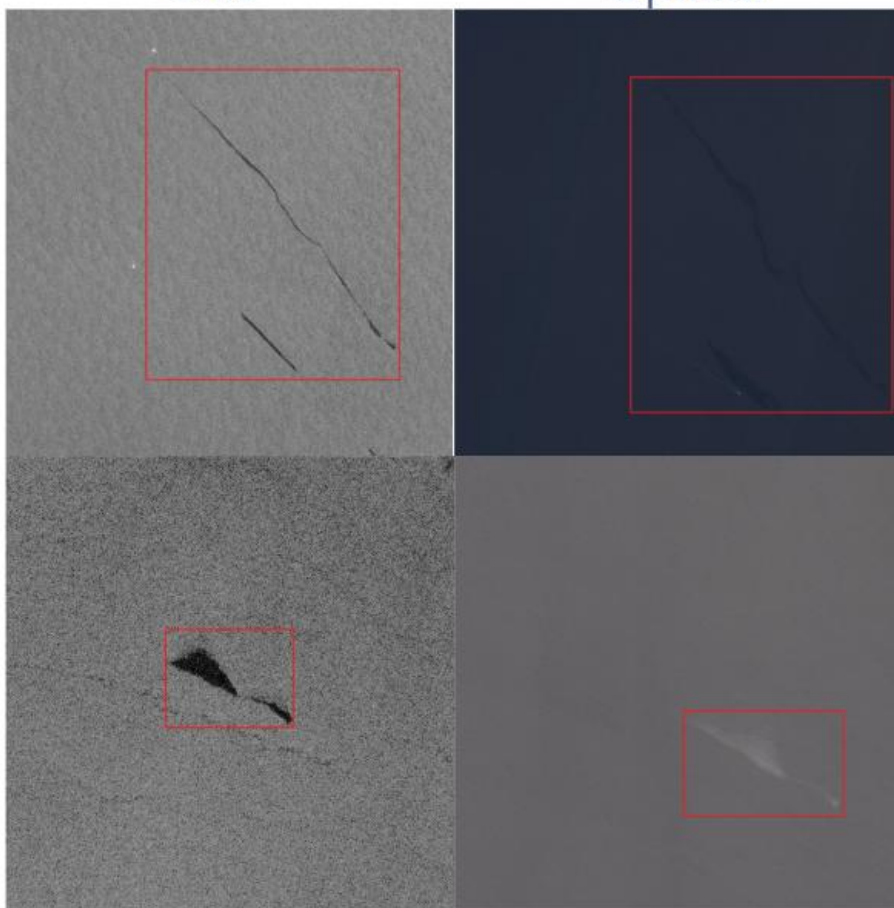
B.1

Test command used

Appendix C. SAR and Optical pair dataset samples

SAR

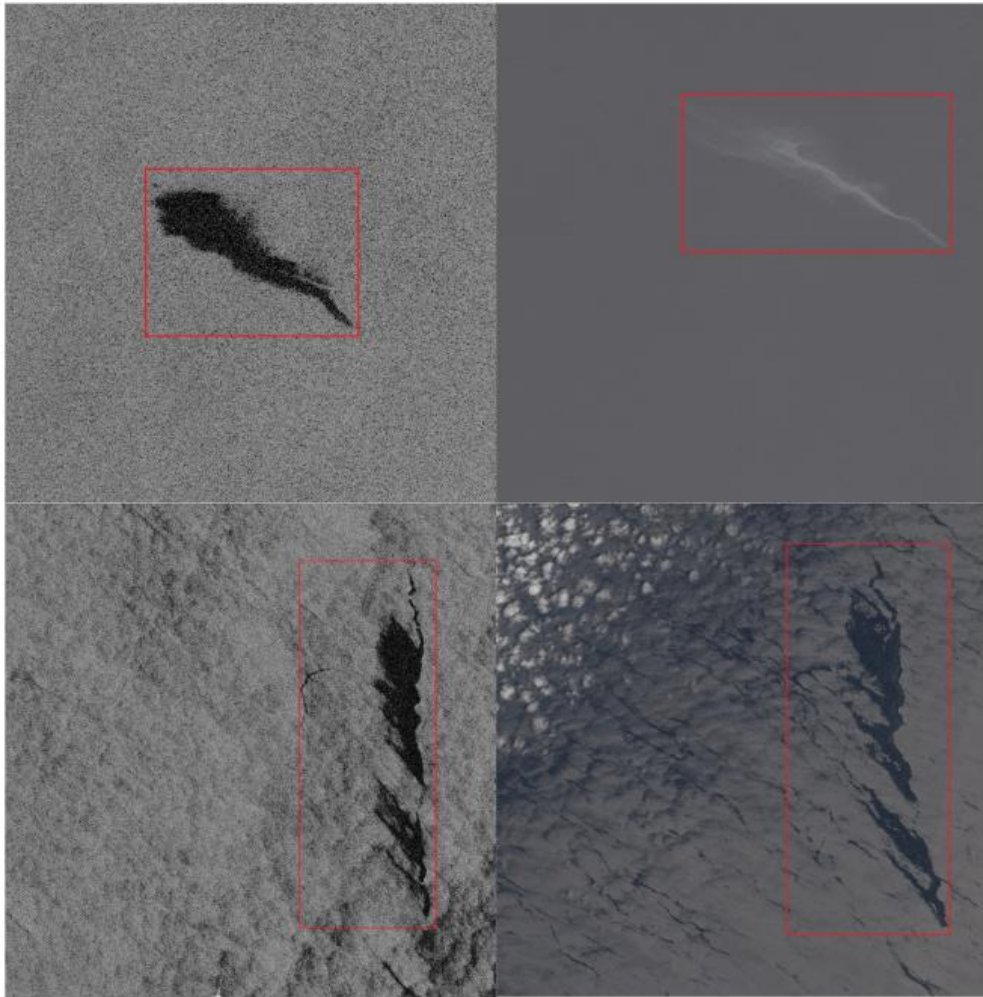
Optical



C.1 pairs 1-2

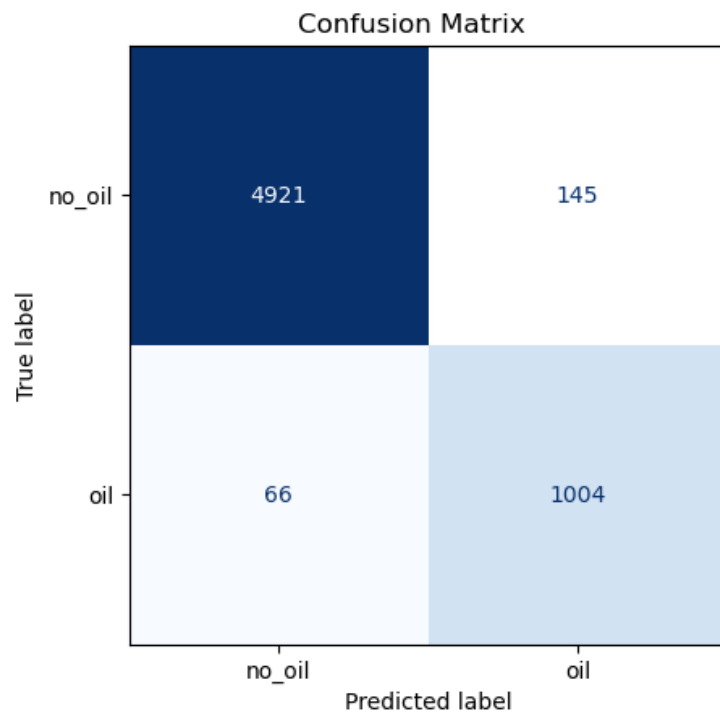
SAR

Optical



C.2 pairs 3-4

Appendix D. Model training results



D.1 Binary classification confusion matrix

Appendix E. Event detection and geocoding:

Detected locations with coordinates						
	location	lat	lon	confidence	severity	title
0	Pujada Bay	6.8535	126.2417	70.0%	Unknown	Coast Guard Confirms No Oil Spill in Pujada Bay
1	East Coast Park	1.305	103.9271	60.0%	Unknown	wild shores of singapore : A restricted East Coast Park site survives the Pasir Panjang oil spill
2	California	36.7015	-118.756	45.0%	Unknown	Trump announces plan for oil drilling off the coasts of California and Florida
3	Trump	39.6823	-76.6022	45.0%	Unknown	Trump announces plan for oil drilling off the coasts of California and Florida
4	Florida	27.7568	-81.464	45.0%	Unknown	Trump announces plan for oil drilling off the coasts of California and Florida
5	San Diego	32.7174	-117.1628	45.0%	Unknown	Trump admin shares plan for oil drilling off California coast - NBC 7 San Diego
6	NBC	55.5655	52.0988	45.0%	Unknown	Trump admin shares plan for oil drilling off California coast - NBC 7 San Diego
7	Miami News	25.7918	-80.2737	45.0%	Unknown	Trump administration announces plan for new oil drilling off the coasts of California and Florida - WSVN
8	Bay Area	37.7885	-122.3558	45.0%	Unknown	Bay Area arts : Climate change and pollution at a UC Davis group show
9	UC Davis	38.5338	-121.7908	45.0%	Unknown	Bay Area arts : Climate change and pollution at a UC Davis group show

E.1 Results of query: event retraction and geocoding.