# Detecting Sea Surface Slicks using Automated Machine Learning

1st Milena Ossenbeck
*CvO University Oldenburg*
Oldenburg, Germany
milena.ossenbeck@uni-oldenburg.de

2nd Daphne Theodorakopoulos
*Marine Perception Research Department*
*German Research Center*
*for Artificial Intelligence (DFKI)*
Oldenburg, Germany
daphne.theodorakopoulos@dfki.de

3rd Jörge Schneemann
*ForWind - Center for wind energy research*
*CvO University Oldenburg*
Institute of Physics
Oldenburg, Germany
j.schneemann@uol.de

4th Oliver Ferdinand
*Marine Perception Research Department*
*German Research Center*
*for Artificial Intelligence (DFKI)*
Oldenburg, Germany
oliver.ferdinand@dfki.de

5th Mariana Ribas-Ribas
*Center for Marine Sensors*
*Institute for Chemistry and Biology*
*of the Marine Environment (ICBM)*
*CvO University Oldenburg*
Wilhemshaven, Germany
mariana.ribas.ribas@uol.de

***Abstract*—Sea surface slicks naturally occur on the ocean. They have important physical, biogeochemical and ecological functionalities. The automatic detection of slicks on images is useful in many scenarios, e.g., to evaluate long time series of images. However, to the best of our knowledge, no methods for the automatic recognition of sea surface slicks on images exist. In this work, a binary classifier is developed that recognises if a slick is present on an image with the help of Automated Machine Learning (AutoML). AutoML automatically finds a machine learning pipeline for a given problem. Two AutoML approaches are compared: auto-sklearn and AutoKeras. Images from seven sites were available from the North and Baltic Seas. Initially, one site was used for training and testing. It was found that the AutoKeras method demonstrated superior performance compared to the auto-sklearn approach in both f1-score and balanced accuracy. After further optimising the model found by AutoKeras and training it on data from the additional sites, the final model reaches an f1-score of 0.924 and a balanced accuracy of 0.954 on the test dataset. Furthermore, the model achieved an f1-score of 0.710 and a balanced accuracy of 0.925 on a holdout dataset only containing images from a site which was excluded from the training. This shows that the model is not only able to reliably recognise sea surface slicks but it also generalises fairly well to unseen data.**

## I. Introduction

Marine sea surface slicks appear when the sea surface microlayer accumulates enough organic matter [1]. Automatically recognising slicks fosters further research. An example of this is estimating the frequency of slicks or relating it to other data, such as wind speed. Furthermore, this is a first step towards knowing when and where a slick will appear. A related problem is the detection of oil spills. For years, machine learning methods have been used to identify oil spills, such as Support Vector Machines [2]. In most cases, the detection of oil spills is conducted using satellite radar images, but in some cases, optical satellite images have also been used [2]. Despite the large number of publications on oil spill detection, little work has been done on automatically recognising sea surface slicks. Recent work by Nichol et al. [3] investigated the detection of various types of slicks, including natural slicks, from sentinel images. However, recognising sea surface slicks from optical images has not been done before.

More specifically, a sea surface slick is a form of sea surface microlayer (SML) [1]. SML is found at the boundary between the ocean and the atmosphere and it consists mainly of organic matter which accumulates at the sea surface [1]. The majority of the molecules forming these SML have - other than the molecules found in oil spills - ambiphilic properties, i. e. one end of the molecule is hydrophobic while the other end is hydrophilic [4, p. 22]. When the concentration of these molecules is high enough, the SML becomes visible which is then called a sea surface slick [1]. Additionally, sea surface slicks have a characteristic ability to dampen capillary waves [4, p. 94]. Figure 1 shows an example of sea surface slicks in the North Sea.

Slicks impact the exchange of gases between the ocean and the atmosphere [5]. Mustaffa et al. [5] studied the dependence of the gas transfer velocity $k_{660}$ on the presence of slicks. Traditionally, $k_{660}$ is parameterised only by wind speed, but other factors are known to affect it directly. They found that in regions with slicks present $k_{660}$ is reduced by 62% compared to regions without slicks. Considering the reported frequency of slicks this leads to a reduction of 7% in the $CO_2$ fluxes in the open ocean and even to a 19% reduction in Norwegian

979-8-3503-3226-1 

Fig. 1. Example image of sea surface slicks. The red boundary indicates the region where the slicks are located.

Fjords [5]. Furthermore, Whitney et al. [6] and Gallardo et al. [7] found that organisms such as zooplankton accumulate at slicks and because of this, slicks are an important feeding spot for fish in their larval stages. Since the survival rate of organisms in their larval stage has a great impact on their species' overall abundance, the frequency of slicks is central to the productivity of the investigated area [6]. At the same time, plastics [7] and other organic pollutants [8] were found to concentrate in slicks.

Due to the physical, biogeochemical and ecological importance of these sea surface slicks it would be interesting to estimate their frequency over long periods of time. Especially for a fixed site, the recording of images at a short interval for a long period of time is feasible. The first step for this is an automatic classifier to distinguish between the presence and the absence of slicks on images of the sea surface.

In this work, two different approaches to building a slick classifier using Automated Machine Learning (AutoML) are presented: auto-sklearn and AutoKeras. AutoML solves the tasks of model selection and hyperparameter tuning and sometimes also feature extraction [9]. Whereas auto-sklearn trains and ensembles classical machine learning models [10], AutoKeras selects and optimises a deep neural network [11]. Auto-sklearn does not work on images directly, thus, features need to be extracted from the images first. All models are trained on a dataset containing images from a period of six months from a fixed site in the North Sea at the offshore wind park Nordergründe. The deep learning approach is also trained and tested on a larger dataset from several sites and seasons.

## II. Related Work

### A. Slicks and Slick Recognition

Romano and Marquet [12] found that in 36% of the images, slicks could be identified at the coast of France over a span of two years. It was found that slicks occur more frequently at lower wind speeds and do not occur at wind speeds higher than 6-7 $ms^{-1}$. Especially at the lowest wind speeds slicks often cover the whole visible area [12]. A later study on the frequency of slicks in the open ocean concluded that slicks only occurred when wind speeds were below 5 $ms^{-1}$ [1]. When the wind speed was below 2 $ms^{-1}$, slicks were always observed. Slick frequency dropped to 11% of the images whereas the recording time was also shorter which might contribute to the lower value compared to the frequency in the coastal area [1]. Romano and Garabetian [13] found a daily cycle in a coastal area where slick frequency decreases first and then increases again as the day progresses.

While Nichol et al. [3] did investigate the detection of natural sea surface slicks amongst other phenomenons, they did not use optical images which are the only type of images available in our work. One of the few studies focusing on optical images to identify oil slicks is the survey by Pan et al. [14]. It focuses on how to calculate features from images which are characteristic of the sea surface roughness. Pan et al. [15] presents more features that can characterise sea surface roughness. Pan et al. [14] and Pan et al. [15] are the basis for the feature extraction in the study at hand. Additionally, features calculated from the grey-level co-occurrence matrix are considered [16, p. 271].

### B. Automated Machine Learning

Automated Machine Learning (AutoML) aims to create a complete pipeline to solve a machine learning task, such as classification, with minimal human intervention [17]. This includes four steps: data preparation, feature engineering, model selection and evaluation. The parameters that are optimised can be parameters affecting the training process (i. e. hyperparameters), model parameters needed for model definition such as the input size for a neural network layer, or data preprocessing parameters like data augmentation [17].

The module auto-sklearn provides a framework for AutoML: given a labelled training and test dataset, a defined loss metric and a computational budget, an ensemble model is built [10, 18]. This model can be comprised of a certain number of base classifiers where the number is controlled by the user. It does not include deep neural networks, only classical machine learning models which are based on statistical methods.

AutoKeras [11] implements AutoML using deep learning: before training, it extracts meta-features from the training data and decides which preprocessing steps need to be done, e.g. encode the labels. Next, the search space is built based on the meta information of the data. That includes the selection of a state-of-the-art neural network and some common hyperparameters. The final step is optimising the chosen network, and the training and preprocessing parameters, such as data augmentation and learning rate. The search algorithm starts from a set of well-performing hyperparameters and always mutates the current best-performing set. AutoKeras selects the best-performing model at the end [11].

## III. Datasets

The following image datasets are used in this work:

1) **Nordergründe**: The dataset which is mainly used in the following contains images taken at the offshore wind park Nordergründe in the North Sea. Coordinates for this site are taken from [19]. From the 23rd of December 2021 until the 2nd of March 2022 images were taken in ten-minute intervals and five-minute intervals after that date until the 30th of June 2022. In total, 23004 images were available in this time frame. The camera used at this site is a LevelOne FCS-4041 dome camera. The images from Nordergründe always show the same perspective except for slight changes in the angle and position.
2) **Beachcamera Spiekeroog**: this dataset contains hourly images taken throughout the year 2021 facing the beach of Spiekeroog. In total, 3186 images are available in this dataset. Coordinates for this site are from [20].
3) **Camera at Lighthouse Alte Weser**: the camera shows parts of the lighthouse platform. Images are available starting from the 12th of May 2022 in five-minute intervals. Images until 30th of June 2022 were considered. In total, 2396 images are available in this dataset. Coordinates for this site are from [21].
4) **Fino1** (Deck and AlphaVentus): research platform in the western North Sea [22]. From this source images from two different perspectives are available: both perspectives show parts of the wind park Alpha Ventus and one shows parts of the platform [22]. For both perspectives, images are taken at ten-minute intervals. The dataset starts on the 8th of March 2022. For both cameras, which in the following will be referred to as Deck and AlphaVentus, images until the 30th of June 2022 were considered which leads to a total of 7884 (Deck) and 7595 (AlphaVentus) images. Coordinates from this site are from [22].
5) **Fino2** (cam02 and cam08): research platform in the Baltic Sea [23]. Here also images from two perspectives are available while both show the wind park Baltic 2 [23]. One also shows parts of the platform (cam08) while the other only shows a small portion of a railing (cam02). The dataset starts on the 8th of March 2022 (cam02) and on the 29th of April 2022 (cam08) respectively and images are taken in a ten-minute interval. For both cameras, images until the 30th of June are considered. For cam02, 7853 images and for cam08 4525 are available in the respective time frames. Coordinates for this site are from [23].

Figure 2 shows where exactly the sites are located in the North and the Baltic Seas. The map was plotted using geopandas [24] and data for the coastlines from [25].

### A. Image Selection

Before labelling, some of the images were discarded when they were blurred or when the wind speed was too high at the time when the picure was taken. Romano [1] showed that slicks do not occur at wind speeds above $5\ m\ s^{-1}$. When wind speed data was available, the images were filtered out if the wind speed above the sea surface exceeded $5\ m\ s^{-1}$. This reduces the imbalance in the data.For the sites, Alte Weser and Nordergründe wind speed data from a sensor at the lighthouse Alte Weser was used. The measurements were taken at a height of approximately $17\,m$ over the sea surface. For Fino2 the database of the Fino-project [26] was used where the heights were given. No, or not enough wind speed data was available for the images from Fino1 and for the images from Spiekeroog. From the logarithmic wind profile [27, p. 76] the relationship in equation 1 can be derived. It was used to calculate the wind speed measurement at which the wind speed above the sea surface was $5ms^{-1}$. Here, $z$ describes the height at which the measurements were taken, $z_r$ is a height right above the sea surface which in this case is set to $0.1\,m$, $u$ is the wind speed at the specified heights and $z_0$ is the characteristic roughness length which was set to $10^{-4}\,m$ [27, p. 77]. When setting $u(z_0)$ to $5ms^{-1}$, the corresponding $u(z_r)$ can be calculated. Additionally, the $u(z_r)$ is rounded up to the next integer. According to this, the dataset is filtered.


Fig. 2. Map of northern Germany and parts of the Baltic and North Seas. The red dots show the locations where the images for the respective datasets were taken.

$$ln\left(\frac{z}{z_0}\right)u(z)^{-1} = ln\left(\frac{z_r}{z_0}\right)u(z_r)^{-1} \tag{1}$$

### B. Labelling the data

All other images were manually labelled as “true” (contains a slick), “false” (no slick) and “unsure” (the labeller was unsure). For the datasets from the sites AlphaVentus and Nordergründe the images labelled as “unsure” were reviewed again and relabelled if a better fitting label could be assigned. An additional label was used for the sites AlphaVentus and Nordergründe which is called ”marginal” and shows small slicks or slicks with undefined edges. Often, these images come before or after images labelled as “true”. In these cases, they show the development or disintegration of slicks. For the training and testing, only the images labelled as “true” or “false” are taken into consideration.

### C. Inter-rater-reliability

To determine the validity of the labelling, 1000 images from the complete dataset are labelled by a second person. For the

subset belonging to the site AlphaVentus, the second labeller was instructed to use the classes “true”, “false” and “marginal” while for the other images, the labeller was supposed to use the classes “true” and “false”. Then, Cohen’s kappa [28] is calculated from the results according to equation 2.

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{2}$$

$P_o$ is the fraction of samples where both labellers agreed on a label and $P_e$ is the fraction of samples where they could have agreed by chance. Instead of just stating how often the raters agreed on a label, Cohen’s kappa also takes into account that an agreement could take place by chance and adjusts the measure accordingly. Cohen’s kappa can take values between $-1$ and $1$, both inclusive, where $-1$ would mean that the raters always disagree and $1$ indicates that both raters always agree. A value of $0$ means that the fraction of samples where the raters agreed correspond to the agreement by chance [28].

For both, the extended dataset as well as the AlphaVentus dataset, Cohen’s kappa was calculated. The values were $0.803$ and $0.747$ respectively. For the AlphaVentus dataset, Cohen’s kappa might be lower because three categories were used instead of two. According to Landis and Koch [29], both these values indicate a substantial agreement which means that the quality of the labelled dataset is satisfactory.

### D. Description of the final datasets

Figure 3 shows the number of images in each dataset after filtering and labelling and the percentage of images labelled as “true” at every site. The dataset size from Nordergründe was clipped to 5000 images to be able to test more features. Each dataset was then split into a subset for training (70%) and for testing (30%). When a validation set was required, the last 20% of the training dataset was used for validation. The dataset Fino1 - AlphaVentus is completely set aside for independent testing.

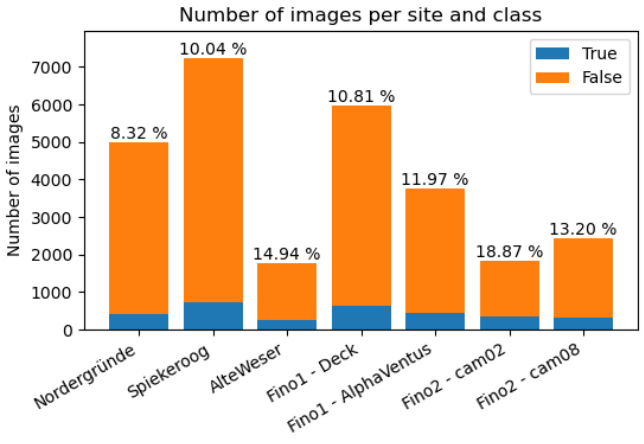


Fig. 3. Number of images per site and class. The percentages denote the portion of slick images per site.

## IV. Methods

The following will first introduce the metrics used to assess the performance of the trained models. Then the feature extraction which is necessary for auto-sklearn is described. AutoKeras performs the feature extraction automatically and can directly take the images as input.

### A. Metrics

Table I introduces the confusion matrix [30, p. 33]. The ground truth value corresponds to the label assigned by hand and the prediction value is the outcome of the model.

TABLE I
The confusion matrix

| | | Prediction | |
|---|---|---|---|
| | | True | False |
| Ground truth | True | true positive (TP) | false negative (FN) |
| | False | false positive (FP) | true negative (TN) |

The dataset is quite imbalanced (cf. Figure 3). In the case of imbalanced datasets the accuracy might distort the actual performance of a model [31]. Therefore, other metrics should be considered for evaluation. In this work, those are balanced accuracy and the f1-score. The balanced accuracy is defined in equation 3 [31]:

$$Balanced\ accuracy = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{3}$$

The first summand of equation 3 is also known as recall which is the correctly predicted portion when looking at all samples labelled as “true” (see equation 4) [32]:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

The correctly predicted portion of all samples classified as “true” by a model is called precision (see equation 5) [32]:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Balanced accuracy does not include precision in its definition. The f1-score combines recall and precision by forming their harmonic mean [32] (see equation 6):

$$f1 - score = \frac{2}{1/P + 1/R} = \frac{2PR}{P + R} = \frac{2TP}{2TP + FN + FP} \tag{6}$$

### B. Feature Extraction

For the approach using auto-sklearn, image features need to be extracted. These features were mainly taken from [14] and [15]. In both papers, the attempt to estimate the roughness of the sea surface was made and as a second step, oil slicks were detected with these features.

*1) Creating the features:* The following features were considered as candidates:

- The **energy** derived from the grey level-gradient co-occurrence matrix [14].
- The **edge frequency** for edge lengths 5, 10, 15, ..., 50 (in pixels) [14].
- The **auto-correlation** function for the maximum offsets 5, 10, 15, ..., 50 (in pixels) [14].
- The **fractal dimension** determined with the **improved box-counting method** [15]. Here, two lengths need to be provided where one is larger than the other. The lengths can take the values 5, 10, 15, ..., 50 and the combinations are created so that the difference between the values can be 5, 10, ..., 40 (in pixels).
- The **fractal dimension** determined with the **grey value statistic method** [15]. Here, the size of the quadratic sub-images is varied from 35, 70, ..., 245, 280 (in pixels).
- From the **grey level co-occurrence matrix** features characterising an image can be derived. This matrix counts how often grey-level pairs occur in an image [16, p. 271]. For this, a fixed distance and a direction to the pixel neighbour need to be defined [16, p. 271]. For every such combination, a new co-occurrence matrix can be created [16, p. 271]. From every co-occurrence matrix several measures are calculated with scikit-image [33]. For these measures, the co-occurrence matrices were calculated for the directions 0° (horizontal) and 90° (vertical) and for each direction the pixels were 1-5 pixels apart. From these ten co-occurrence matrices, the following features are computed:
  - homogeneity
  - energy
  - contrast
  - dissimilarity
  - correlation
- Furthermore, on the base of 40 co-occurrence matrices with directions 0°, 45° and 90°, 135° and pixel distances 1 - 10 **weighted co-occurrence matrices** are created [15]. Specifically, for every pixel distance, the four co-occurrence matrices are averaged, where each one corresponds to a different direction, according to Table 1 in [15]. For every feature that is supposed to be extracted, the weights are different. The following features are computed using the implementations of scikit-image [33]:
  - homogeneity
  - angular second momentum
  - contrast
  - entropy

The image features are extracted from the grey-scale images with their original size of 1920 x 1080. Subsequently, the image features of the training data are standardised and then, using the same mean and standard deviation, the test set features are standardised.

*2) Selecting the features:* After the features are created, a subset of them is selected to increase efficiency and because a classifier learns relevant patterns better [30, p. 266 ff.]. Several different methods exist to reduce the number of features. First, features are selected using the t-test and after that two filter methods are considered. Filter methods discard features only based on their values and interactions but do not consider the classifier [30, p. 266 ff.].

Some of the features were filtered out using the one-sample t-test [34, p. 273 f.]. For that, all described features were extracted from 100 randomly selected images. This was repeated for the same images with the difference that 10% of pixels in the images were ignored randomly. This procedure was followed to ensure the validity of the features for another approach which was compared to the approach in this work. Then, for every feature, a ratio was calculated by comparing the two features extracted from both image versions. After that, the t-test was carried out on the distribution of the ratio for every feature. Features were excluded when the t-test's p-value is smaller than 0.05.

Subsequently, the features are filtered by the mutual information criterion (MIC) or by the f-value. Both of these tests are implemented in scikit-learn [35]. For an explanation of the MIC, refer to Zhou [30, p. 267 f.]. The f-value is calculated based on ANOVA (Analysis of variance) [36, p. 146]. The analysis is supposed to answer the question of whether two distributions are different or whether they are subsets of the same underlying distribution [30, p. 146]. While the f-value detects linear dependencies, the MIC also detects other dependencies between features [37]. Both criteria are used independently to filter features in the following.

To avoid choosing too many similar features, per feature only the best variation as defined by the list in IV-B is selected based on the f-value or the MIC. Only for the features calculated from the single grey-level co-occurrence matrices, the best variation per direction (0°or 90°) is selected. For example, one feature is chosen from the contrast calculated from the co-occurrence matrices where the pixel distances are 1-5 and the direction between the pixels is 0°.

## V. Experiments and Results

The experiments performed and their results are described below. The first part describes how auto-sklearn was applied to the selected image features. The second part describes how AutoKeras was applied to the images directly. Next, additional hyperparameter tuning was applied to the final model selected and trained by AutoKeras. Finally, the model is trained on the extended dataset containing images from six different sites.

### *A. Slick Recognition based on Image Features using auto-sklearn*

*1) Experimental Setup:* The AutoSklearn Classifier [10] is applied to the features selected as described in section IV-B. Before classification, oversampling was applied to the set of features calculated from the images because of its imbalanced distribution of classes. The SMOTE technique was used for this [38]. The following settings and hyperparameters were altered and tested:

- **Feature selection method**: MIC or f-value (f).
- **Number of base classifiers in the ensemble**: 25 or 50.
- **Optimisation objective**: accuracy or f1-score.

In total, auto-sklearn was trained for eight combinations. For every trial, an hour of computation time was allocated.

*2) Results:* The results are reported in table II. The best model was optimised towards the f1-score, consisting of 50 models and achieves an f1-score of 0.658 on the test dataset. The features for this model were selected using the f-value. The features used in this model are the following:

- The energy derived from the grey level-gradient co-occurrence matrix;
- The edge frequency for edge length 10;
- The auto-correlation function for the maximum offset 5;
- The fractal dimension determined with the improved box-counting method with the length combination 10 and 15;
- The fractal dimension determined with the grey value statistic method with the sub-image size 280.
- For the features derived from the single co-occurrence matrices, the following combinations were chosen (pixel distance, direction):
  - homogeneity (5, 90°), (5, 0°)
  - energy (5, 90°), (5, 0°)
  - contrast (5, 90°), (4, 0°)
  - dissimilarity (5, 90°), (5, 0°)
  - correlation (5, 90°), (5, 0°)
- For the features derived from the weighted co-occurrence matrices, the following features were chosen (pixel distance):
  - homogeneity (10)
  - angular second momentum (10)
  - contrast (10)
  - entropy (10)

TABLE II
RESULTS OF ALL EIGHT CONFIGURATIONS USING AUTO-SKLEARN

| filter | objective | models | f1-score | balanced accuracy |
|---|---|---|---|---|
| f | f1-score | 50 | **0.658** | **0.817** |
| f | f1-score | 25 | 0.653 | 0.816 |
| MIC | accuracy | 50 | 0.648 | 0.815 |
| MIC | f1-score | 50 | 0.644 | 0.808 |
| f | accuracy | 25 | 0.638 | 0.800 |
| f | accuracy | 50 | 0.621 | 0.774 |
| MIC | accuracy | 25 | 0.621 | 0.788 |
| MIC | f1-score | 25 | 0.613 | 0.773 |

### B. Slick Recognition with AutoKeras

*1) Experimental Setup:* For the Deep Learning approach, the ImageClassifier class of AutoKeras was used [11]. Before using the grey-scale images for training and testing they are resized to 224 x 224. The parameters were set as follows:

- **Batch size**: 32 (16 for large models)
- **Number of trials**: 50
- **Loss function**: binary cross-entropy
- **Optimisation objective**: f1-score

When more complex models were tested by AutoKeras, the batch size was automatically reduced to 16 because of limited memory. During the training, the last 20% of the training dataset was used as the validation dataset. To avoid overfitting, an early stopping condition was introduced: when the validation f1-score does not improve by more than one percentage point for ten epochs, the training is stopped. In the end, the model that reached the highest f1-score was chosen. Furthermore, a threshold of when an image is classified as "true" was determined manually after training by maximising the f1-score on the training data.

*2) Results:* The best architecture chosen by AutoKeras was EfficentNet-b7 with an f1-score of 0.842 and a balanced accuracy of 0.887 when setting the threshold to 0.566. It is the most complex version of EfficientNet [39]. This model was trained with a batch size of 16. The final hyperparameters selected by AutoKeras are listed in table III.

TABLE III
LIST OF THE HYPERPARAMETERS OF THE BEST MODEL CHOSEN BY AUTOKERAS WHEN USING THE PREDEFINED SEARCH SPACE

| Hyperparameter | chosen value |
|---|---|
| normalize | True |
| augment | True |
| translation_factor | 0.1 |
| horizontal_flip | False |
| vertical_flip | False |
| rotation_factor | 0 |
| zoom_factor | 0 |
| contrast_factor | 0 |
| block_type | efficient |
| pre-trained | True |
| trainable | True |
| version | b7 |
| imagenet_size | True |
| reduction_type | global_avg |
| dropout | 0 |
| optimiser | adam |
| learning_rate | 2e-5 |

Figure 4 shows the precision-recall curve of the test (red) and training data (black).

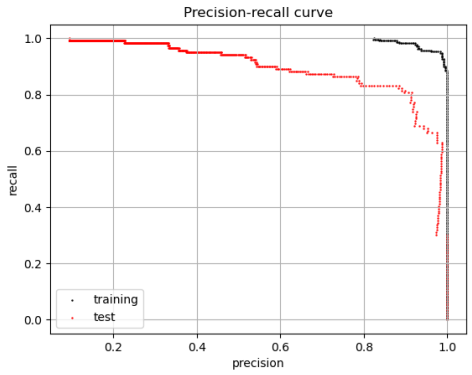


Fig. 4. Precision and recall at varying thresholds for the test and train data.

### C. Hyperparameter tuning of EfficientNet

AutoML methods try many configurations of the machine learning pipeline. Since the budget, e.g. the number of trials, is often limited, not all options were tested and the best option might not have been found. Therefore, additional hyperparameter tuning is performed on the final model. For this, the selected model architecture, i.e. EfficientNet-b7, was retrained starting from the weights of the ImageNet dataset [40] which was also the starting point for the AutoKeras search [11]. Only the learning rate was tuned.

*1) Experimental Setup:* The Hyperband-Tuner is used [41] with the Hyperband implementation of the module KerasTuner [42]. Since $2 * 10^{-5}$ is the learning rate of the model returned by the AutoKeras search, the new values are centred around this value. The ten tested learning rates are logarithmically spaced between $10^{-6}$ and $10^{-4}$. The performance of a model is determined by the f1-score on the validation dataset. The maximum number of epochs was set to ten to achieve this within a reasonable timeframe.

*2) Results:* The best-performing learning rate was $2.783 * 10^{-6}$. The threshold maximising the f1-score on the training data is 0.5. Table IV summarises the results of all three methods, namely the auto-sklearn approach, the model found by AutoKeras, and the found AutoKeras architecture retrained with the tuned learning rate $\eta$. While the f1-score on the test dataset has only risen by $0.002$ to $0.844$, the balanced accuracy has improved by $0.027$ to $0.914$. This shows that the additional hyperparameter tuning actually improved performance.

TABLE IV
RESULTS OF ALL THREE METHODS

| Method | Threshold | f1-score | balanced accuracy |
|---|---|---|---|
| Auto-sklearn | 0.5 | 0.658 | 0.817 |
| AutoKeras | 0.566 | 0.842 | 0.887 |
| AutoKeras + tuned $\eta$ | 0.5 | **0.844** | **0.914** |

### D. Training the model on the extended dataset

*1) Experimental Setup:* To be able to recognise slicks at different sites, the model needs to be trained on the extended dataset which contains images from six different sites. The sample size of the extended dataset contains 14530 samples (70%) in the training and 6230 samples (30%) in the test subset which makes it around four times larger than the previously used dataset. In the following, the obtained model is trained on this extended dataset and its performance is evaluated on the extended test set as well as an unseen seventh dataset, the AlphaVentus dataset. The training starts from scratch with the architecture and hyperparameters obtained in the previous section. Early-stopping is applied: when the loss on the validation dataset has not decreased for five epochs, the training is stopped and the model from five epochs prior is restored. The training on the extended dataset concluded after 22 epochs and weights of epoch 17 were restored.

*2) Results:* Figure 5 shows the precision-recall curves for this model on the training (black curve) and the test dataset (red curve). The threshold is set to $0.636$ which maximises the f1-score on the training dataset. As expected, the model performs slightly better on the training than on the test dataset which indicates that no or minimal overfitting is taking place. The precision-recall curve for the test dataset suggests a good overall performance for the obtained model.

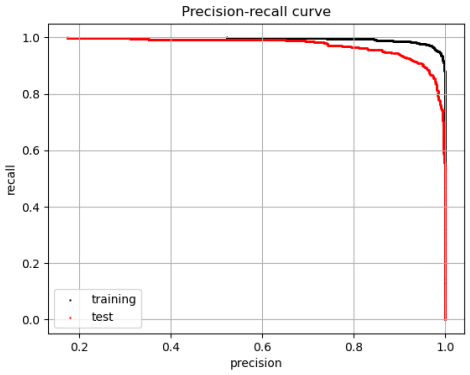


Fig. 5. Model trained on the extended training dataset for 17 epochs. The precision-recall curves for the prediction of the model on the training as well as the test dataset are shown.

Table V shows the performance of the model trained on the Nordergründe dataset only (single site) and the model trained on the extended dataset (six sites). Both are evaluated on the Nordergründe dataset, the extended test dataset and the unseen AlphaVentus dataset. Both models perform similarly on the Nordergründe dataset. The overall f1-score achieved on the extended test dataset by the new model is $0.924$ and the balanced accuracy is $0.954$. Not surprisingly, the performance on the completely unseen dataset is decreased compared to familiar sites. Whereas the single-site model performed very poorly on both unseen datasets, the six sites model was able to generalise fairly well to the AlphaVentus dataset.

TABLE V
PERFORMANCES OF THE MODEL TRAINED ON A SINGLE SITE VS SIX SITES ON THE TEST SET, THE EXTENDED TEST SET AND THE UNSEEN ALPHAVENTUS DATASET

| training | threshold | evaluation | f1-score | balanced accuracy |
|---|---|---|---|---|
| single site | 0.5 | test dataset | **0.844** | 0.914 |
| six sites | 0.636 | test dataset | 0.841 | **0.924** |
| single site | 0.5 | extended test dataset | 0.316 | 0.660 |
| six sites | 0.636 | extended test dataset | **0.924** | **0.954** |
| single site | 0.5 | AlphaVentus (unseen) | 0.381 | 0.525 |
| six sites | 0.636 | AlphaVentus (unseen) | **0.710** | **0.925** |

## VI. DISCUSSION

This section will analyse why AutoKeras outperforms auto-sklearn. We then consider the generalisation ability of the final model. Finally, we will comment on the dataset.

### A. Model Comparison

The results show that deep learning with AutoKeras clearly outperforms the feature extraction approach combined with auto-sklearn on the task of recognising marine surface slicks on images. The best auto-sklearn model achieved an f1-score of $0.658$ and a balanced accuracy of $0.817$. The AutoKeras model is $0.184$ higher in the f1-score and $0.083$ better in the balanced accuracy.

One explanation for the overall moderate performance of the auto-sklearn approach is that the selected features do not capture all of the relevant information. Either the features might be unsuitable for the task or the features are still not diverse enough meaning that many of them capture similar information. To see whether the models improve, more features could be added. However, because deep neural networks automatically learn suitable features during the training process, it seemingly produces a better outcome.

Another explanation could be that the training might have been cut-off early or the hyperparameters were not set well. Increasing the maximum duration and testing more hyperparameter settings might yield better results. This is also true for the AutoKeras approach, more trials make it possible to test more configurations. Moreover, the subsequent hyperparameter tuning for the AutoKeras approach could be more extensive by considering more hyperparameters and more values.

### B. Generalisation Ability of the Model

Section V-D examines how the model performs when it is trained on the extended dataset and tested on an unseen dataset. The performances of the model trained on the images from Nordergründe on the extended test dataset as well as the AlphaVentus dataset are reported in table V. While overall the model trained on the extended dataset shows an improvement in both f1-score ($0.924$) and balanced accuracy ($0.954$) on the extended test dataset compared to the model trained on the smaller dataset, it performs similarly to the previous model on the images from Nordergründe. This indicates that the model trained on the extended dataset did not suffer from underfitting which might happen if the training data is too diverse so that no patterns can be learned by the model.

The performances on the extended test dataset and the AlphaVentus dataset have both improved strongly compared to the single-site model: the difference in performance is 0.4 for the balanced accuracy and 0.329 for the f1-score on the unseen AlphaVentus dataset. This shows that training on more diverse images leads to the classifier learning more general aspects of a slick's appearance. The fairly good performance on the unseen dataset shows the generalisation ability of the model. However, the f1-score and the balanced accuracy on the AlphaVentus dataset still stay behind the performance on the extended test dataset. This is expected since no images from AlphaVentus were seen during training.

### C. Datasets and Labelling

In this work, four classes were considered for the labelling of the datasets (true, false, marginal, unsure), of which only images of the two labels "true" and "false" were used. The inter-rater reliability showed that the labelling can be considered reliable.

The class "true" contains both images where the whole sea surface is covered with a slick and images where the slicks are fragmented. However, these cases might be quite different in the recognition process. Labelling different strengths of a slick and seeing it as either a multi-class classification problem or a regression problem might improve performance. This might especially improve the auto-sklearn approach since the features used for the classical machine learning models focus on texture. For example, in one case the whole sea surface is smooth while in the other case, the presence of edges might indicate a slick. For this, more data needs to be available to get a sufficient amount of images where the whole sea surface is covered in a slick.

In general, the number of slick images in the dataset is small. Considering that the distribution of images among the sites is different, there is only a small number of slick images from some sites.

## VII. Conclusion and Future Work

In this work, a model was developed that can detect the presence of a sea surface slick in an optical image. Two Automated Machine Learning methods were compared: auto-sklearn with self-engineered image features and AutoKeras using deep learning. AutoKeras clearly outperformed auto-sklearn by 0.184 in the f1-score and 0.08 in balanced accuracy. AutoKeras found EfficientNet-b7 to be the best choice for identifying sea surface slicks. The learning rate for the model was fine-tuned after the search process which marginally improved performance. The final model was retrained on more data (six different sites instead of one). On the extended test dataset, an f1-score of $0.924$ and a balanced accuracy of $0.954$ are achieved. Moreover, the model's ability to generalise to images from unseen sites was investigated by measuring the performance on an unseen dataset. Its performance only moderately decreased compared to the performance on familiar sites: here, the f1-score drops to $0.710$ and the balanced accuracy reaches a value of $0.925$. This is a huge difference from the model only trained on data from one site which failed to recognise slicks on images from unknown sites reliably. This shows that the model trained on six sites is able to identify marine sea surface slicks at several different, also unfamiliar, locations with a relatively high recall and precision.

From this, the following future work can be derived:

1) **Improve the method**. To improve the auto-sklearn method, more features could be added. The AutoKeras method might find a better model when the budget is increased. Moreover, more extensive hyperparameter tuning can be done. It is also possible that another AutoML method performs better.
2) **Larger, more diverse dataset**. A larger, more diverse training dataset would improve the robustness of the model. That includes data from more sites, different seasons of the year and different angles of the camera.

Also, splitting the "true" label into "whole slick" and "fragmented slick" might improve performance.

3) **Slick segmentation**. Another next step could be the development of a model localising the slick in the image and estimating the area covered by it. This might be important information when estimating the impact of sea surface slicks on a process such as the gas transfer velocity in an area.
4) **Slick frequency**. The model could help in estimating the general frequency of slicks at a large scale.
5) **Slick prediction**. Given other data, such as wind speed or sea surface temperature, the conditions that might or might not favour the development or disintegration of sea surface slicks could be analysed. Based on that, a model predicting when and where a slick will occur could be developed.

## ACKNOWLEDGMENTS

The images used in this study from the sites Nordergründe and Alte Weser were taken within a measurement campaign supported by the German Federal Ministry for Economic Affairs and Climate Action on the basis of a decision by the German Bundestag (WindRamp, grant no. 03EE3027A). We want to thank the operator of the offshore wind farm Nordergründe "Skyborn Renewables offshore solutions GmbH" and the operator of the light house Alte Weser "Wasserstraßen- und Schifffahrtsamt Weser-Jade-Nordsee" for their support of the measurement campaign in WindRamp which made this work possible in the first place. Furthermore, the wind speed data for the location Fino2 was provided by the FINO project organized by Projektträger Jülich, supported by the German Federal Ministry for Economic Affairs and Climate Action and coordinated by the Bundesamt für Seeschifffahrt und Hydrographie. This work was also funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 451574234

## REFERENCES

[1] J.-C. Romano, "Sea-surface slick occurence in the open sea (Mediterrrean, Red Sea, Indian Ocean) in relation to wind speed," *Deep-Sea Research I*, vol. 43, no. 4, 1996.

[2] R. N. Vasconcelos, A. T. C. Lima, C. A. D. Lentini, G. V. Miranda, L. F. Mendonça, M. A. Silva, E. C. B. Cambuí, J. M. Lopes, and M. J. Porsani, "Oil Spill Detection and Mapping: A 50-Year Bibliometric Analysis," *Remote sensing (Basel, Switzerland)*, vol. 12, no. 21, p. 3647, 2020.

[3] J. E. Nichol, A. S. Antonarakis, and M. Nazeer, "Monitoring the sea surface microlayer (sml) on sentinel images," *Science of The Total Environment*, vol. 872, p. 162218, May 2023.

[4] M. Gade, *Marine Surface Films: Chemical Characteristics, Influence on Air-Sea Interactions and Remote Sensing*. Springer-Verlag Berlin-Heidelberg, 2006.

[5] N. I. H. Mustaffa, M. Ribas-Ribas, H. M. Banko-Kubis, and O. Wurl, "Global reduction of in situ $CO_2$ transfer velocity by natural surfactants in the sea-surface microlayer," *Proceedings of the Royal Society. A, Mathematical, physical, and engineering sciences*, vol. 476, no. 2234, p. 20190763, 2020.

[6] J. L. Whitney, J. M. Gove, M. A. McManus, K. A. Smith, J. Lecky, P. Neubauer, J. E. Phipps, E. A. Contreras, D. R. Kobayashi, and G. P. Asner, "Surface slicks are pelagic nurseries for diverse ocean fauna," *Scientific reports*, vol. 11, no. 1, pp. 3197–3197, 2021.

[7] C. Gallardo, N. C. Ory, M. d. l. Gallardo, M. Ramos, L. Bravo, and M. Thiel, "Sea-Surface Slicks and Their Effect on the Concentration of Plastics and Zooplankton in the Coastal Waters of Rapa Nui (Easter Island)," *Frontiers in Marine Science*, vol. 8, 2021.

[8] F. Garabetian, J.-C. Romano, R. Paul, and J.-C. Sigoillot, "Organic matter composition and pollutant enrichment of sea surface microlayer inside and outside slicks," *Marine environmental research*, vol. 35, no. 4, pp. 323–339, 1993.

[9] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.

[10] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-Sklearn 2.0: Hands-free automl via meta-learning," vol. 23, no. 261, pp. 1–61, 2022.

[11] H. Jin, Q. Song, and X. Hu, "Auto-Keras: An Efficient Neural Architecture Search System," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 1946–1956.

[12] J.-C. Romano and R. Marquet, "Occurrence frequencies of sea-surface slicks at long and short time-scales in relation to wind speed," *Estuarine, coastal and shelf science*, vol. 33, no. 5, pp. 445–458, 1991.

[13] J.-C. Romano and F. Garabetian, "Photographic Records of Sea-Surface Microlayers as a survey of pollution daily rythm in coastal waters," *Marine Environmental Research*, vol. 41, no. 3, 1996.

[14] H. Pan, P. Gao, H. Zhou, R. Ma, J. Yang, and X. Zhang, "Roughness analysis of sea surface from visible images by texture," *IEEE access*, vol. 8, p. 1, 2020.

[15] H. Pan, W. Zhang, W. Jiang, P. Wang, J. Yang, and X. Zhang, "Roughness Change Analysis of Sea Surface From Visible Images by Fractals," *IEEE access*, vol. 8, pp. 78 519–78 529, 2020.

[16] A. Distante and C. Distante, *Handbook of Image Processing and Computer Vision: Volume 3: from Pattern to Object*. Cham: Springer International Publishing AG, 2020.

[17] F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019, available for free at http://automl.org/book.

[18] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information*

*Processing Systems 28 (2015)*, 2015, pp. 2962–2970.

[19] Bundesministerium für Wirtschaft und Klimaschutz, "Nordergründe," n.d., accessed on 28.11.2022. [Online]. Available: https://www.erneuerbare-energien.de/EE/Redaktion/DE/Standardartikel/Offshore-Windenergie/Projeke/nordergruende.html

[20] ICBM Universität Oldenburg, "Informationen," n.d., accessed on 28.11.2022. [Online]. Available: https://uol.de/icbm/forschungsplattformen-schiffe/messstation/informationen

[21] P. Menz, "Längengrad, Breitengrad, GPS-Koordinaten von Leuchtturm Alte Weser," n.d., accessed on 28.11.2022. [Online]. Available: https://www.laengengrad -breiten-grad.de/gps-koordinaten-von-leuchtturm-alte-weser

[22] FINO1, "Standort," n.d., accessed on 28.11.2022. [Online]. Available: https://www.fino1.de/de/standort.html

[23] DNV, "Geographische Lage," n.d., accessed on 28.11.2022. [Online]. Available: https://www.fino2.de/de/projekt/standort.html

[24] K. Jordahl, J. V. den Bossche, M. Fleischmann, J. Wasserman, J. McBride, J. Gerard, J. Tratner, M. Perry, A. G. Badaracco, C. Farmer, G. A. Hjelle, A. D. Snow, M. Cochran, S. Gillies, L. Culbertson, M. Bartos, N. Eubank, maxalbert, A. Bilogur, S. Rey, C. Ren, D. Arribas-Bel, L. Wasser, L. J. Wolf, M. Journois, J. Wilson, A. Greenhall, C. Holdgraf, Filipe, and F. Leblanc, "geopandas/geopandas: v0.8.1," Jul. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3946761

[25] GADM, "Download GADM data (version 3.6)," n.d., accessed on 28.11.2022. [Online]. Available: https://gadm.org/download_country_v3.html

[26] Bundesamt für Seeschifffahrt und Hydrographie, "FINO - Login FINO-Datenbank," accessed on 17.12.2022. [Online]. Available: http://fino.bsh.de/

[27] H. Kraus, *Grundlagen der Grenzschicht-Meteorologie: Einführung in die Physik der Atmosphärischen Grenzschicht und in die Mikrometeorologie*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[28] K. Pykes, "Cohen's Kappa," 2020, accessed on 02.12.2022. [Online]. Available: https://towards-datascience.com/cohens-kappa-9786ceceab58

[29] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[30] Z.-H. Zhou, *Machine Learning*, 1st ed., ser. Springer eBook Collection. Springer Nature Singapore Pte Ltd., 2021.

[31] Scikit-learn developers, "Metrics and scoring: Quantifying the quality of predictions," n.d. a, accessed on 12.05.2022. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html

[32] D. J. Hand, P. Christen, and N. Kirielle, "F: an interpretable transformation of the F-measure," *Machine learning*, vol. 110, no. 3, pp. 451–456, 2021.

[33] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014. [Online]. Available: https://doi.org/10.7717/peerj.453

[34] K. Koutroumbas and S. Theodoridis, *Pattern recognition*, 4th ed. Elsevier Inc., 2010.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[36] D. C. Hoaglin, F. Mosteller, and J. W. Tukey, Eds., *Fundamentals of exploratory analysis of variance*, ser. Wiley series in probability and mathematical statistics. John Wiley Sons, Inc., 2008.

[37] Scikit-learn developers,, "Comparision of F-test and mutual information," n.d. b, accessed on 08.11.2022. [Online]. Available: https://scikit-learn.org/stable/auto_examples/feature_selection/plot_f_test_vs_mi.html

[38] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, jun 2002.

[39] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[41] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018. [Online]. Available: http://jmlr.org/papers/v18/16-558.html

[42] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "KerasTuner," 2019. [Online]. Available: https://github.com/keras-team/keras-tuner