Computer technologies to determine offshore facilities suitable for the climatic conditions

I Shatrovskii¹ *, A Zolotukhin²

¹ Northern (Arctic) Federal University
² Russian State Gubkin University of Oil and Gas, Russia

* Contact author: i.v.shatrovskii@gmail.com

Abstract. This article provides an overview of computer technologies and an analysis of their application possibilities to determine the quantitative and qualitative composition of the facilities for the development of offshore oil and gas fields for given climatic conditions. The first question covered in the article is how to form a data set including the information about offshore fields development conditions. The author gives examples of using the Python programming language, NumPy, Tesseract and OpenCV libraries for that purpose. For each designated variable, a test program code was done to obtain the required value. Another question, which is given in the article, is how to analyze the obtained results and determine which offshore facilities are best suited for certain climatic conditions. For this purpose, a deep neural network of forwarding propagation was created with the help of Keras, TensorFlow and Scikit-learn Python libraries. The network performance was tested on a sample of a small size, the results were analyzed, and future research tasks were defined.

1. Introduction

Taking into account large accumulated base of theoretical knowledge in the field of artificial intelligence, successful pilot projects of leading oil and gas companies in this area, the increased interest in recent years in the problems of implementation of artificial intelligence technology in the oil and gas industry [1] and the fact that, according to many projections, the global world demand for oil in the next 20 years will increase by 20%, it can be concluded that research in this area is currently very relevant and can potentially have a significant positive effect for the development of the oil and gas sector.

This paper aims to find possible solutions to the problems of atomization, the processes of collecting, preparing and processing information about the climatic conditions of the location of offshore oil and gas fields and the subsea and surface infrastructure used for their development. This information can be used to design developing a new offshore project in the early stages within the synthetic assessment approach [2].

The main objectives of the paper are: the creation of a data set sample including information about offshore fields; designing a neural network to analyze the correlation between the characteristics of the location of fields and the infrastructure used for their development; application of a neural network to predict the concept of field development that is not included in the data set sample; analysis of results.
2. Creating a data set sample

Data collection for analysis was carried out on the example of the British North Sea shelf using a public National Data Repository (NDR) launched by the UK Oil and Gas Authorities in 2019 [3].

The sample includes such basic parameters as:
- name of a field;
- field type (oil, gas or gas condensate);
- sea depth in the area, in meters;
- distance to shore, in kilometres;
- the area of a field, in square kilometres;
- number of elements of surface infrastructure for production;
- number of elements of subsea infrastructure for production;
- average distance between elements of the subsea and surface infrastructure of the field, in kilometres.

The variables listed above can be manually captured using the NDR's user interface tools in the Map section (on a visually displayed map).

Since the entire process of collecting data can be carried out by a person analyzing the image (map), the author considers that this task can be automated and significantly accelerated by means of computer vision methods, which is known as a scientific area in the field of artificial intelligence and related technologies for obtaining images of real-world objects, processing them and using the obtained data for solving various kinds of problems without the participation of humans (full or partial) [4].

To analyze the ability of such technologies to do this work, we have to test it on a simple task with simple images. To create some of these, the author used the visualization tools of the NDR: in the "Infrastructure" tab, the display marker in the "Surface Inf", "Sub Surface Inf", and "Pipelines" lines are ticked; in the "Fields" tab, a display marker and a label marker in the "Fields" line are ticked; in the "General" tab, the display marker in the "Coastline" line is ticked; in the remaining tabs, prohibit the information appearing as shown on Figure 1.

As a result, a typical fragment of the map looks as shown in Figure 2, where the colored bounded areas are fields (red - gas, purple - condensate, green - oil); blue dots - underwater infrastructure; brown dots - surface infrastructure; lines of all colors are pipelines, and the limited yellow areas of a larger size (compared to fields) are onshore. The field name will also be displayed.
For image analysis, the Python [5] programming language, NumPy [6], Tesseract [7] and OpenCV [8] libraries were used, which are the most widely used computer vision libraries, including hundreds of image processing functions and are used both in academic institutions, and in IT industry. Importing libraries in Python is as follows:

```python
import numpy as np
import pytesseract
import cv2
```

To obtain a value of a variable responsible for the name of a field, the most effective and simple solution, in the authors' opinion, is to use the Tesseract library for recognizing text in an image loaded with OpenCV:

```python
# Tesseract location
pytesseract.pytesseract.tesseract_cmd = 'C:\Program Files\Tesseract-OCR\tesseract.exe'

# Loading an image and converting it to RGB
img = cv2.imread('field_name.png')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Showing the text from an image
config = r'--oem 3 --psm 6'
print(pytesseract.image_to_string(img, config=config))
cv2.waitKey(0)
```
Thus, for example, for the image "field_name.png" shown on Figure 3, the program returns the value "CALDER".

![Figure 3. Image «field_name.png»](image.png)

The value of the field type variable can be obtained by returning the value of the color encoding that prevails within the field contour. In Python, using OpenCV, this can be implemented as:

```python
# Load an image and convert it to grayscale
im = cv2.imread('field_name.png')
gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
# Contours detection
contours,h=cv2.findContours(gray,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
# Creating an array "mask" to determine the average color value inside the contour
final = np.zeros(im.shape,np.uint8)
mask = np.zeros(gray.shape,np.uint8)
for i in xrange(0,len(contours)):
    mask[...,]=0
    cv2.drawContours(mask,contours,i,255,-1)
```

Since only one color is inside the contour, the operation returns average values that are the same for the entire area and characterize the shade of red with the parameters Red: 254, Green: 192, Blue: 192 for a gas image (see Figure 3), a shade of purple with parameters Red: 241, Green: 219, Blue: 254 for a condensate field and a shade of green with parameters Red: 206, Green: 254, Blue: 188 for an oil field.

The value of the variable characterizing the depth of the sea in the area of the field can be determined by comparing (in the correct scale) the coordinates of the contours centers of the fields on the image (map) with the coordinates of the depths, which can be obtained using the user interface of the NDR in the section "Data Search ". The program code for displaying the coordinates of the contour centers ("cnts" variable) looks like this:
# Creating an array containing the coordinates
coordinates = []

# Finding contours
cnts=cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if len(cnts) == 2 else cnts[1]

# Calculation of centroids
for c in cnts:
    area = cv2.contourArea(c)
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.05 * peri, True)
    M = cv2.moments(c)
    cx = int(M['m10']/M['m00'])
    cy = int(M['m01']/M['m00'])

# Filling the array containing the coordinates
coordinates.append((cx, cy))

# Displaying the array containing the coordinates
print(coordinates)
cv2.waitKey()

The distance to the shore and the average distance between the elements of the subsea and surface infrastructure of the field can be determined quite simply using standard Python tools. So, because we know the coordinates of two points on the map A (x1, y1) and B (x2, y2), then to find the distance, we can use the hypot function, which returns the Euclidean norm:

```python
from math import hypot
distance = hypot(x2 - x1, y2 - y1)
```

Determination of the area of the field using OpenCV is implemented after finding the contours (variable cnt) by determining the moments of the image as follows:

```python
# Determination of the moment
M = cv2.moments(cnt)
# Determination of the centroid coordinates
cx = int(M['m10']/M['m00'])
cy = int(M['m01']/M['m00'])
# Determination of the area within a contour
area = cv2.contourArea(cnt)
```

The number of elements of surface and subsea infrastructure for production can be calculated by limiting by area the contours recognized in the image because the points reflecting the elements of the infrastructure are much smaller than the areas that represent the fields. In this case, the program code for the finding contours (cnts) will look like this:

```python
# Limiting area conditions
s1= 3
s2 = 20
xcnts = []
for cnt in cnts:
    if s1<cv2.contourArea(cnt) <s2:
        xcnts.append(cnt)
# Displaying the number of infrastructure elements
print("infrastructure number: {}".format(len(xcnts)))
```
To store all received data an *.xlsx file was generated containing records of fifty fields with various schemes of infrastructure development for production. Finally, the main factors influencing the type and quantity of installed equipment were identified and designated as variables:

- Name - the name of a field (for reference);
- Type - field type (Oil - oil, Gas - gas, Cond - gas condensate);
- Depth (m) - sea depth in the area of the field, in meters;
- Distance (km) - distance to the shore, in kilometers;
- Square (km2) - field area, in square kilometers);
- Surfinf - number of elements of surface infrastructure for production;
- Subsurfinf - number of subsea infrastructure elements for production;
- AvDist (km) - average distance between elements of the subsea and surface infrastructure of the field, in kilometers.

For the fifty fields mentioned above, the file contains the values of all variables. If the field does not have surface infrastructure, then Surfinf parameter is set to "0"; if the field does not have subsea infrastructure, then the Subsurfinf parameter is set to "0"; if one of the Surfinf or Subsurfinf parameters, or both of these parameters are set to "0", then the AvDist (km) parameter is also set to "0".

3. Designing a neural network

Since the neural network will face the task of predicting values based on existing data, the most appropriate choice in terms of architecture will be a deep feedforward neural network, which is based on the backpropagation method.

To create a neural network in Python [5], the libraries Keras [9], TensorFlow [10], as well as Scikit-learn [11] were used to normalize the data and divide the sample into training and test sets. To create graphs, the Matplotlib library was used [12].

Based on the task, on the input layer of the neural network, we have 4 neurons for the parameters Type, Depth (m), distance (km), Square (km2), then there are 5 fully connected layers, consisting of 16 neurons with the activation function ReLU, selected experimentally. On the last layer, we have 3 neurons, which reflect the parameters we need Surfinf, Subsurfinf and AvDist (km).

Since the variable Type has a non-numeric value, for the neural network to work, it had to be recoded using standard Python tools. After that, the stage of normalizing all values of random variables was performed using the StandardScaler Scikit-learn tool. The sample was divided into 67% of the training data and 33% of the test data, and the mean square error value was chosen as the loss metric.

4. Results

During the training of the neural network, the optimal number of training epochs was set equal to 320, at which the final value of the loss function on the tested data was the smallest and amounted to 30, which indicates a weak correlation between the input and output data. The graphs of the values of the loss functions in the training and test samples (loss and val_loss, respectively) at different epochs are presented on Figure 4.
5. Future developments

According to the author's opinion, described developments can be used as elements of the interpreter and knowledge base editor for an expert system accumulating the knowledge of specialists in the field of offshore field development design and replicating this empirical experience for consulting less qualified users or to speed up the design process.

According to the problem being solved, an expert system will belong to the class of systems for solving the problem of data interpretation. Interpretation refers to the process of defining the meaning of the data, the results of which must be consistent and correct. Typically, multivariate data analysis is used.

By connection with real time, the system will be classified as static, since the data on development concepts of offshore structures will not change over time, but only be updated with new information.

The type of computer used to work with such an expert system should not be more complicated than a personal computer (IBM PC), so that the system can be used both for research and educational purposes, and in production.

In terms of the integration with other programs, the system should be autonomous, since the task that it will face is quite specific and narrowly focused, while the design and use of a hybrid expert system is more complex task than the development of an autonomous one. Docking not just different packages, but different methodologies (what happens in hybrid systems) creates a whole range of theoretical and practical difficulties.

Thus, the future system can be implemented through the already used Python programming language and additional modules and libraries. The user interface - a component of programs that implement the user dialogue with the system at the stages of entering information, achieving results and "explaining" the solution can be designed using a set of PyQt extensions.

The knowledge base - the core of the system, which is a collection of formalized knowledge about the existing structures of the infrastructure for production in a form understandable to an expert and a knowledge engineer, can be designed in the same way as described in paragraph 2.

The neural network, the development of which is described in the article, should act as the basis for the interpreter - a program that simulates the course of the expert's reasoning, and new program modules should be developed for the explanations subsystem and the knowledge base editor. At least four people will be involved in the development and operation: an expert, a knowledge engineer, a programmer and a user.
6. Conclusion
The analysis showed that all the tasks required for automation of the collection and preparation of data on the climatic conditions of fields and on the infrastructure used for their development could be solved using computer vision methods. Still, processing of this information does not give the desired result with the described model of data and neural network. During the analysis of the results, the author identified a number of reasons leading to a weak correlation of parameters, the main of which are the insufficient sample size and the fact that the fields selected for the experiment were chosen randomly, without taking into account the surrounding infrastructure of nearest fields and the availability to connect with onshore equipment. As a direction for future research, the authors set the following tasks:

- designing a parser for automated creation of a larger sample;
- solving the clustering problem for the obtained sample. The fields need to be grouped around major nodes of existing transportation infrastructure;
- solving the regression problem within each cluster.

Acknowledgment: The authors acknowledge the support from HR.dr UTFORSK Project (project number UTF-2017-four-year/10044) between University of Stavanger and Gubkin Russian State University of Oil and Gas.

References
[1] Exploring the impact of artificial intelligence on offshore oil and gas. / OffshoreTechnology/ URL: https://www.offshore-technology.com/features/application-of-artificial-intelligence-in-oil-and-gas-industry/ (Accessed 1 August 2021).
[2] O Gudmestad, Zolotukhin, E Jarlsby. Petroleum Resources With Emphasis on Offshore Fields. P 288.
[3] UK NDR URL: https://ndr.ogauthority.co.uk/, (Accessed 1 August 2021).
[4] Jan Erik Solem. Programming Computer Vision with Python: Tools and algorithms for analyzing images. P 314.
[5] Python URL: https://www.python.org/, (Accessed 1 August 2021).
[6] NumPy URL: https://numpy.org/, (Accessed 1 August 2021).
[7] Tesseract URL: https://github.com/tesseract-ocr/tesseract/, (Accessed 1 August 2021).
[8] OpenCV URL: https://opencv.org/, (Accessed 1 August 2021).
[9] Keras URL: https://keras.io/, (Accessed 1 August 2021).
[10] TensorFlow URL: https://www.tensorflow.org/, (Accessed 1 August 2021).
[11] Scikit-learn URL: https://scikit-learn.org/, (Accessed 1 August 2021).
[12] Matplotlib URL: https://matplotlib.org/, (Accessed 1 August 2021).