Sergey Parkhonyuk

temp3

Обновлено 21 августа 2024, 08:54

Challenge context

Provider description

SLB is a global technology firm established in France in 1927 by Conrad and Marcel Schlumberger, initially focusing on cabling services for the oil sector. Since 1929, significant investments in research and strategic acquisitions have enabled SLB to become a leader in its industry. Currently, the company operates in over 100 countries and employs individuals from nearly 200 nationalities, striving daily to promote energy innovation for a sustainable planet. Their efforts center on advancing technology in the oil and gas industry, implementing large-scale digital solutions, decarbonizing various sectors, and fostering the development of new energy sources to accelerate the energy transition.

Context

Corrosion-related failures pose significant hazards and account for more than 25% of overall well failures in the oil and gas industry (Figure 1). Steel pipe corrosion over time due to extreme wellbore conditions can lead to adverse financial consequences and potential environmental impacts, such as groundwater contamination, gas leakage, and surface seepage. Pipe condition evaluation for well integrity workflows involves corrosion characterization, leak and structural deficiency monitoring, and detection of potential pipe breaks. Accurate corrosion detection is crucial for both operators and service companies to reduce interpretation time, minimize subjectivity in monitoring, and enhance overall performance.
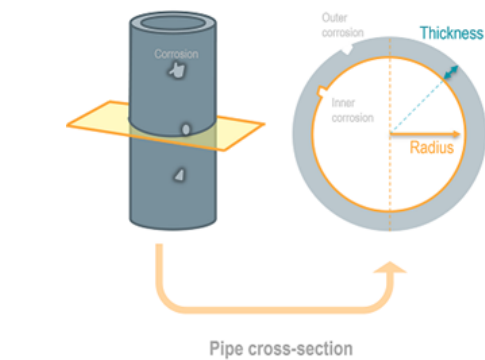

Figure 1 - Corrosion visualization on the well.

To inspect steel pipes and detect corrosion along the wellbore, various types of logging imagers are utilized, including electro-magnetic, ultrasonic, and mechanical imagers. These imagers provide detailed topographic maps of pipe walls or thicknesses, containing information about the pipe's current state, such as manufacturing process, collar positions, and defects in the inner or outer wall.

Received pipe thickness images are mappings of the cylindrical-pipe dimensions in polar coordinates. These maps are viewed as 2D images (y-axis being the depth and x-axis the azimuth). THBK represents the variation of the thickness around the mean thickness value THAV. The specificity of this data is that it is long (wells can be up to kilometers) and narrow (azimuthal resolution is limited). Due to telemetry, some processing errors might occur, corrupting data points on the maps. Therefore, appropriate data cleaning and processing are required before analyzing the data and using them for training as grayscale images (Figure 2).
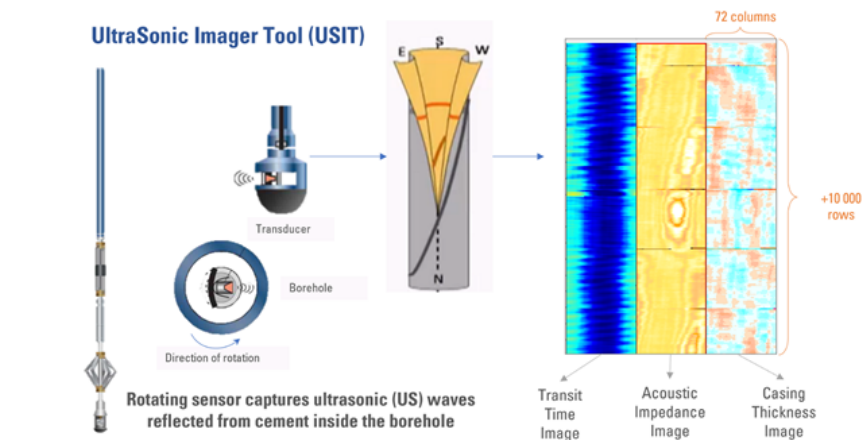

Figure 2 - Ultrasonic Imager Tool; tool sensors and the images obtained.

Glossary

- **Drill pipe** is a hollow, thick-walled piping used in drilling to transmit drilling fluid and torque to the drill bit.

- **THBK :** Thickness Variation - The variation of the pipe's thickness around its average value.

- **THAV:** Average Thickness - The mean thickness value of the pipe.

- **Azimuth:** Horizontal axis of the images, representing the angular measurement in a spherical coordinate system, indicating the horizontal angle relative to a cardinal direction, typically north.

Challenge goals

The goal of the hackathon is to produce a model that gives the highest possible score for groove defect segmentation.

This competition is evaluated on the intersection over union (IoU). The IoU can be used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. The formula is given by:

$$\frac{|X \cap Y|}{|X \cup Y|}\frac{|X \cap Y|}{|X \cup Y|}$$

where XX is the predicted set of pixels and YY is the ground truth. The IoU is defined to be 1 when both XX and YY are empty. The leaderboard score is the mean of the IoU coefficients for each image in the test set.

Data description

Data Description

The pipes manufactured using a mold represent manufacturing patterns. Manufacturing patterns are homogenous within small sections of the pipes called joints; they are organized in repetitive and visually coherent shapes and forms (circles and horizontal, vertical, or oblique lines with varying thickness). However, intra joint feature distribution is high; two consecutive joints within the same well can have distinctly different manufacturing patterns. These patterns reflect loss and excess of metal due to the manufacturing process.
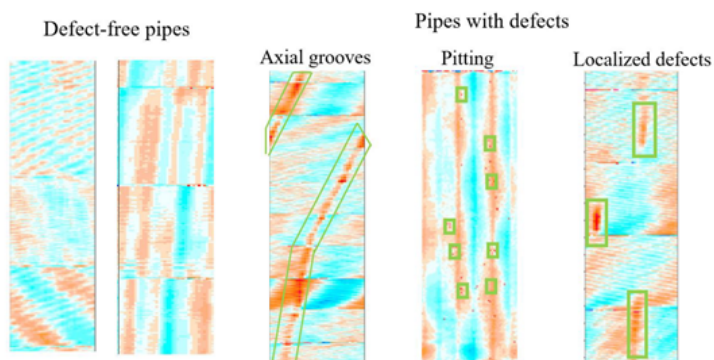

Figure 3. Corrosion and wear defects Examples

Collars are defined as the junction between two joints and are displayed in the radius and thickness map as a line.

Manufacturing patterns and collars represent a background on which defects and anomalies are overlaid. Corrosion in the pipe always appears as metal loss (red patterns in below figure); pipes lose their thickness in the inner or in the outer wall, sometimes simultaneously. (Corrosion is not an either-or process; it may happen on both the inner and outer wall at the same time). Those defects are random in size, metal penetration, and overlay on top of the manufacturing patterns.

Corrosion and wear defects on steel pipes can be classified into three categories:

• Pitting corrosion

• Localized defects

• Axial groove

As mentioned earlier, the database comprises ultrasonic images from 20 wells. Each image within the database is paired with a binary mask of identical dimensions, acting as the corresponding label for that specific image. The well dimensions are outlined as follows:

Train & Validation data:

- Well 1: 3008 x 72 – vertical groove

- Well 2: 1088 x 36 – vertical discontinuous groove

- Well 3: 63892 x 36 – vertical discontinuous groove

- Well 4: 2136 x 36 – vertical groove

- Well 5: 1774 x 72 - vertical groove

- Well 6: 17364 x 72 – diagonal groove

- Well 7: 29800 x 72 - vertical and diagonal groove

- Well 8: 1424 x 72 – diagonal groove

- Well 9: 3016 x 72 – no groove

- Well 10: 1470 x 72 – no groove

- Well 11: 54457 x 36 – vertical and diagonal groove

- Well 12: 1593 x 36 - vertical groove

- Well 13: 68580 x 36 – vertical discontinuous groove

- Well 14: 12800 x 36 – diagonal groove

- Well 15: 7328 x 36 – diagonal groove

The images being really big, we already cut them into smaller square patches of size 36x36 pixels. Each image is saved using the following name convention: `well_<well_id>_patch_<patch_id>.npy` You can load them by using the `np.load` function.

The labels are binary images of size 36x36, they are saved in a csv file. In order to load them you should read the csv and reshape the labels into the right format.

```
# Read file
y_train=pd.read_csv(Path('y_train.csv'), index_col=0) #Table with index being the name of the patch

# Access to one patch label
y_train.loc['well_1_patch_130'].reshape(36*36)

# Get all labels at once
y_train.values.reshape((-1,36,36))
```

Format of the output

The format for the output is a csv where each line is the flatten patch. Here is an example for create the csv file from saved prediction.

```
from tqdm.auto import tqdm

img_save_dir = Path(f'../data/predictions')
for img_path in tqdm(img_save_dir.glob('*.npy')):
    name = img_path.stem
    if name in labels[phase]:
        continue
    label = np.load(img_path)
    label_tsh = (label>0.5)*1
    labels[phase].update({name:label.flatten()})

pd.DataFrame(labels['Test'], dtype='int').T.to_csv(Path(f'../data/pred.csv'))
```

Benchmark description

The benchmark is established on a simple CNN architecture, meticulously trained on 36x36 patches extracted from the well dataset. At the end of the architecture we added a sigmoid function.

Pre-processing operations were executed to enhance data quality:

- Missing values were replaced with zeros.

- Patches containing outlier values were systematically eliminated from the training set.

- Robust Scaler normalization was individually applied to each patch for consistent scaling.

The hyperparameters for training were carefully configured:

- The CNN architecture consists of 5 layers without pooling.

- A batch size of 128 was chosen to optimize computational efficiency during the training process.

- The learning rate was set at 0.001 to guide the model through effective convergence.

- Training was conducted over 30 epochs to capture the temporal evolution of features within the data.

- Utilizing the Binary Cross Entropy loss function facilitated effective optimization, striking a balance between the dice coefficient and the binary cross-entropy components.

- Data augmentation techniques, including flip and horizontal roll, were strategically incorporated to enhance the model's adaptability.

- The optimizer used during training was the Adam optimizer.

In contrast, no specific post-processing steps were applied, maintaining the integrity of the model's output.