1. Describe the problems inferred from looking at some sample images.

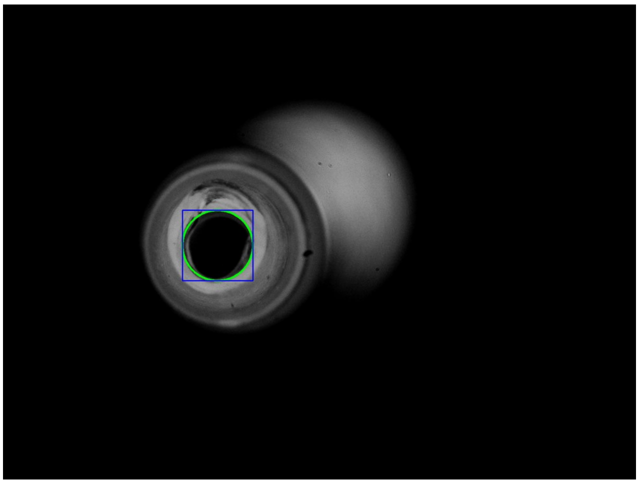
Determining the end of the nail/metal with good production quality. Out of 24 presented images, there are 3 categories, namely OK, ARGUABLY GOOD, and NO GO. Only the categories of OK and ARGUABLY_GOOD are worthy of being sold.

2. Propose solution(s) to the problem found above, describe it in short paragraph/diagram/pseudo.

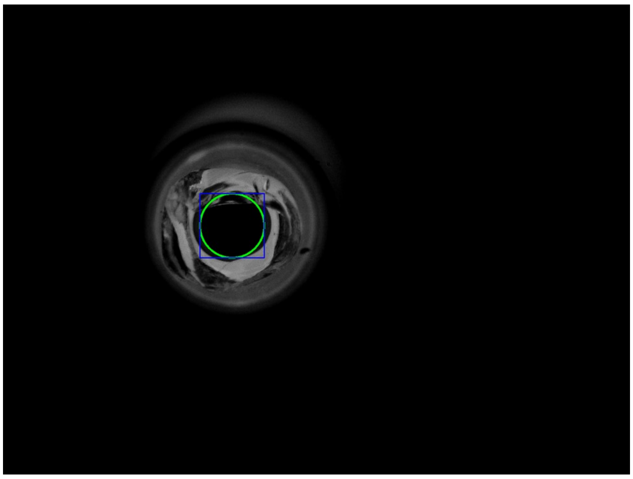
```
Import required libraries and modules

Load pre-trained VGG16 model
    Modify the VGG16 model by removing the last layers responsible for classification
    Set the model to evaluation mode

Define function load_image(image_path):
    Open the image at the given path using PIL
    Resize the image to (224, 224) pixels
    Return the resized image

Define function get_image_embeddings(object_image, model):
    Define a preprocessing pipeline for the image
    Preprocess the object image using the defined pipeline
    Add an additional dimension to the image tensor
    Pass the preprocessed image through the model to obtain an image embedding
    Return the image embedding

Define function get_similarity_score(first_image, second_image):
    Calculate image embeddings for the first and second images using get_image_embeddings function
    Calculate cosine similarity between the two embeddings
    Return the similarity score

Main:
    Define the paths for reference images (OK and ARGUABLY_GOOD) and the directory containing images to compare
    Load the reference images and store them as tensors using load_image function

    Iterate through each image in the image directory using glob:
        Load the current image using load_image function

        Calculate similarity score between the current image and the reference image using get_similarity_score function
        Calculate a threshold score using the reference images

        If similarity score is greater than or equal to the threshold score:
            Set status as "OK"
        Else:
            Set status as "NOT-OK"

        Load and process the current image using OpenCV
        If status is "OK":
            Save the processed image in the OK directory
        Else:
            Save the processed image in the NOT-OK directory

        Resize and display the processed image using OpenCV, along with status and similarity score
        Print the similarity score

Exit
```

Result

ARGUABLY GOOD | 0.98



NO GO | 0.86



3. code:

berikut langkah – langkah menjalankan aplikasi.

1. unzip formulatrix.zip
2. cd /formulatrix
3. source virtual/bin/activate
4. python3 inference.py

#info: *Development aplikasi dibangun diatas Operating System Ubuntu*