

Time

- Please complete 5 days from the time you receive this email with the test instructions.

Our fellow colleague, Bjorn is swamped with tedious manual work and needs our help! You see Bjorn has been assigned to figure out if N pairs of images are different. For the past 4 months he has been manually opening each pair and giving them each a "Bjorn score". Tired and frustrated Bjorn is asking us to automate this process.

Bjorn will always ensure that you will be given a CSV file with N pairs (example provided below); additionally, he states that:

- The CSV will contain 2 fields (image1 and image2) with N records.
- Each field contains the absolute path to an image file.

image1 image2

```
aa.png ba.png
ab.png bb.png
ac.png ac.gif
ad.png bd.png
```

The tool should output the result in a CSV file with the following requirements:

- The CSV file will need to have 4 fields (image1, image2, similar, elapsed) and have the same amount of records as the input file.
- The values that fall under the first 2 fields (image1 and image2) need to be the same as the input file.
- The values that fall under the similar field will need to represent a "score" based on how similar image1 is to image2. Bjorn is entrusting you to figure out an appropriate scoring algorithm, although he is requesting that '0' indicates that the pair are the same image. **NOTE:** Bjorn needs you to **compare the visual appearance** of each image, **not the binary contents** of each pair of image files (ie. the same image provided in different formats, say, png and gif, should have a score of '0').
- Your manager, Jeanie, would like to calculate the cost of running this task. She is asking you to include the time it takes to compute each score, in seconds, in the output CSV 'elapsed' field.

The output file should look like:

image1 image2 similar elapsed

```
aa.png ba.png 0 0.006
ab.png bb.png 0.23 0.843
ac.png ac.gif 0 1.43
```

image1 image2 similar elapsed

ad.png bd.png 1 2.32

While Bjorn uses MacOS at work, he would also love to run this tool on his windows computer when he works remotely from his home.

Important Considerations:

- How do you know if your code works?
- How are you going to teach Bjorn how to use the program?
- Your manager Jeanie is assigning you to a different task and is making Ferris the maintainer of your application. How do you make sure he succeeds?
- How are you ensuring Bjorn gets the latest version of your application?

Deliverables:

- Return the project folder in as a Gitlab / GitHub / Bitbucket repo.
- Include instructions on how to run various components of the project, as well as how to install any required libraries.
- Please provide a descriptive project README (Markdown format preferred), presenting the various steps you approached while designing and implementing the solution. Some diagrams would be nice, but not mandatory.

Time

- 5 days from the time the candidate receives the email with the test instructions.

You can contact us via email for any questions or clarifications. We will try to respond promptly.