Model documentation and write-up

1. Who are you (mini-bio) and what do you do professionally? If you are on a team, please complete this block for each member of the team.
    a. Stepan Konev: I work as a Machine Learning Engineer developing recommender systems at scale. I used to work as a researcher-developer developing motion prediction module for self-driving cars and delivery robots.
    b. Yuriy Biktairov: I am a computer science PhD student at the University of Southern California. My research primarily focuses on neural network verification techniques and motion prediction for autonomy.

2. What motivated you to compete in this challenge?
    a. Stepan Konev: I am excited with autonomy technology and had a successful experience in participating a similar competition. However, this competition was additionally exciting with it's severe limitations for inference time that sounds like a challenge I wanted to take
    b. Yuriy Biktairov: The problem seemed quite challenging, so it was interesting to give it a try.

3. High level summary of your approach: what did you do and why?

    Our solution is based on classic computer vision techniques and includes 3 major steps. First, we match visual features between a given target image and the base image (the first one). Given a set of matches, we recover the relative pose using a variation of the RANSAC algorithm. Finally, we validate the resulting pose and fallback to a heuristic prediction in case the reconstructed result is deemed unrealistic.

4. Do you have any useful charts, graphs, or visualizations from the process? -

5. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.

6. Please provide the machine specs and time you used to run your model.
    - CPU (model): AMD Ryzen
    - GPU (model or N/A): N/A
    - Memory (GB): 4GB
    - OS: Ubuntu 22.04.4 LTS
    - Train duration: N/A
    - Inference duration: ~10 s / trajectory

7. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)? -

8. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission? -

9. How did you evaluate performance of the model other than the provided metric, if at all?
    a. We used a provided module for metric computation for offline evaluation.
    b. We also evaluated different feature matching approaches in terms of the total number of matches and the average matching confidence.

10. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?
    a. Deep learning based dense feature matching.
    b. Depth estimation of different portions of the satellite.
    c. Some of the more advanced variations of the RANSAC-inspired approaches.
    d. Recovering plain surfaces of the spaceship by color clustering.

11. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?
    a. Perhaps I would implement a specific depth estimation model and utilize the info about surfaces in RANSAC.

12. What simplifications could be made to run your solution faster without sacrificing significant accuracy?
    a. Feature extraction is currently the bottleneck. It could possibly be tuned to perform more efficiently without losing any quality.