1) Project title and summary

Title: Photography techniques.
Summary: The main project idea is to mimic photographic techniques with still images, implemented in a dashboard with sliders for parameter adjustment and showing part of the algorithm used.

2) Project Pipeline and baseline results

A) Literature review: Select in the literature what photographic techniques can be mimic with computer vision techniques. Some ideas are panning, High Dynamic Range (HDR) imaging, depth of field manipulation.
B) Implementation: Implement a photographic technique using python in google colab, using image processing libraries like OpenCV or Pillow.
C) Dashboard Creation: Show the results in a dashboard created using Streamlit.
D) Parameter Adjustment: add a slider to the dashboard to change a parameter and see the effect in the output image.
E) Algorithm Visualization: Show parts of the algorithms in the same dashboard and how it values changed by changing the values in the slider. For example, If the algorithm uses kernels, then we can show the matrix and give the option of changing the values in that matrix and see the response in the output image.
F) Algorithm Improvement: Implement other computer vision algorithms that improve the output image, such as adaptive blurring, edge detection or depth estimation, that probably will require the use of deep learning frameworks such as PyTorch or TensorFlow.

   Baseline: The minimum results was a dashboard showing the results of a photographic technique over an image, in this case a car and blurred background suggesting the car was moving.

3) How you plan to improve on baseline results.

   I will use segmentation algorithms, segmentation by neuronal networks or other tools to segment better the fix objects.

4) Documentation of your experiments and results

   We tried some segment algorithms such as Felzenswald, active contour, edge, cluster or region-base segmentation, but the results were not the ones we wanted. One problem

was that the color of the car was similar to the color of the street and also similar to some background. We also see problems isolating the car by edges because the background image has trees that also are detected by the edges.

Then we tested using neural networks, but we failed in making them work.

Finally, we used another approach and instead of using algorithms to segment an object, we searched and used algorithms/libraries to remove the background. We used rembg and we successfully isolated the car, but the borders are a little wider, as you can see in the tires of the car. We also only managed to isolate the car, and not it shadow.

Original Image	Isolated car using rembg	Example of blur effect

Therefore, we still need to isolate the shadow of the car and some borders around the tires of the car.

The dashboard was created using the libraries OpenCV, Numpy, Streamlit and pandas. The code is at https://github.com/Francisco-hub-eng/cis5810_final/tree/master, and the dashboard is at https://cis5810final.streamlit.app/ . In the notebook car_segmentation.ipynb you can find all the segmentation algorithms tested and its results.

5) Project timeline, milestone, and how the duties for each group member is organized.

I will work alone, and the project timeline is:
- two weeks to implement the neural network algorithm in my computer or other approach for a better segmentation of the car and its shadow.
- one week to implement the segmentation in the dashboard.

Milestone are:
- run a neuronal network in my computer or other method.

- obtain an acceptable segmentation (at least follow the shape of the object, not necessary the borders).
- implement the segmentation in the dashboard.