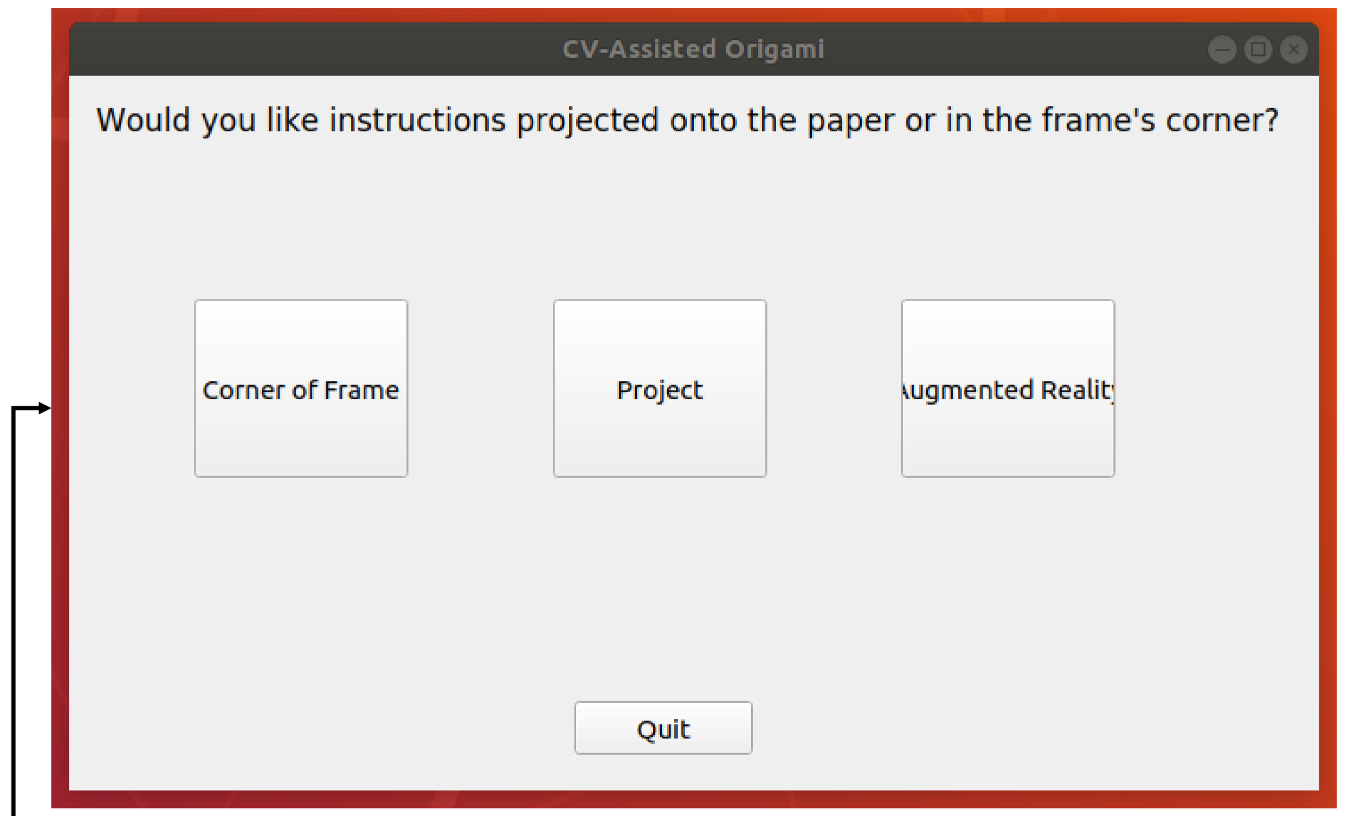
Computer-Vision-Assisted Origami

Our Graphical User Interface allows the user to choose from the three instruction types our program offers and origami pieces the user wants to make.

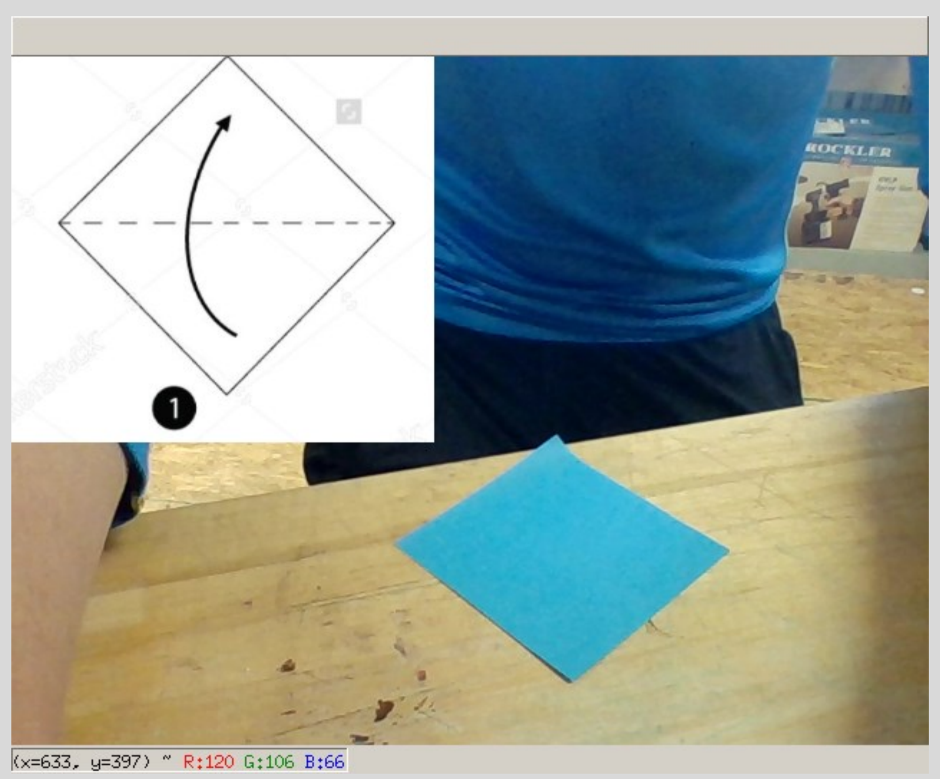


Overview

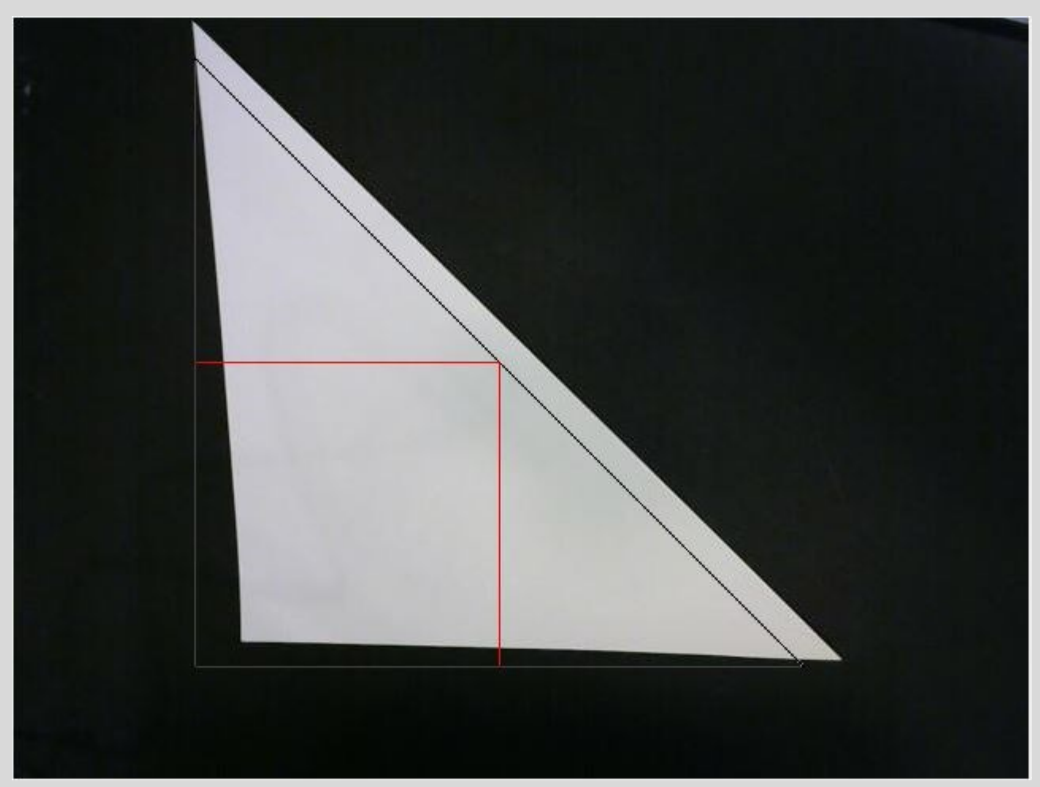
Our group began with two interests: machine learning and computer vision. However, as the intersection between the two is either already done or the subject of PhD research, we began to ask what we enjoyed: origami. Origami instructions are often difficult to translate onto the paper in your own hands. Therefore, we sought to fix this problem by projecting the instructions onto your piece of paper on a display. To explore the solution, we first placed the instruction image on the corner of the live video feed. Once this was finished, we placed and scaled the instruction onto the video feed by comparing area and centroids of shapes. The final implementation was to use a more robust paper detection heuristic and then use a technique called homography to translate the instructions onto the detected paper, regardless of rotation.

The Challenge

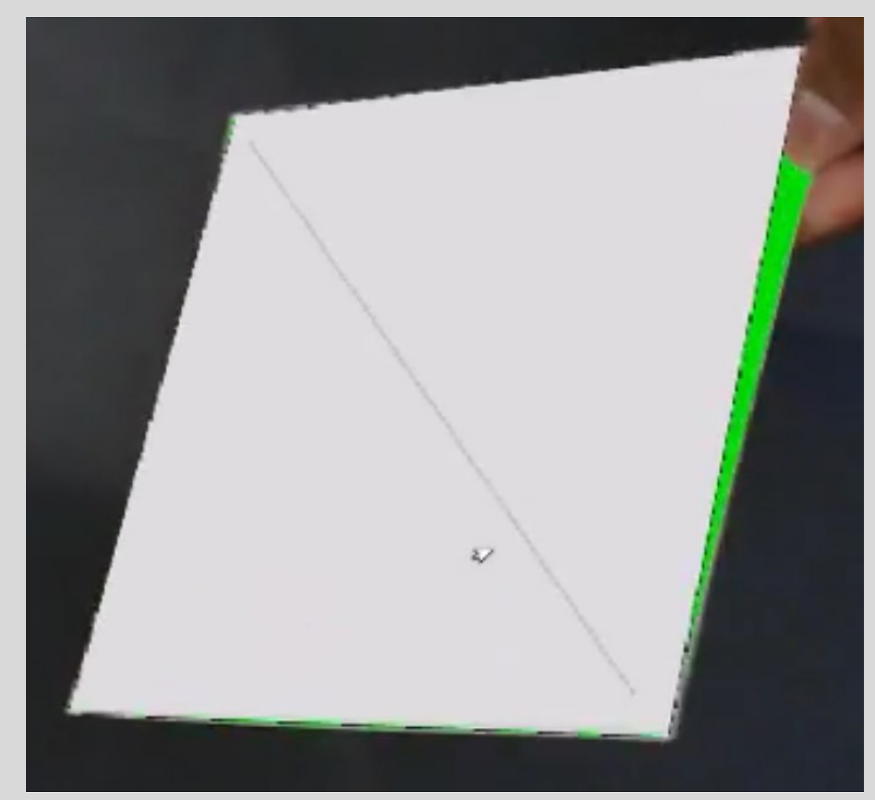
To assist people with folding origami, we first identified two major challenges inherent to the process: interpretation of instructions, and execution of folds. Individuals who aspire to fold up their own origami creations however, probably wish to execute the folds involved of their own volition, and so we determined the most useful challenge in the space worth addressing dealt directly with the clarification of the fold explanations involved in any particular instruction set.



The first iteration of the technology used in this project involved displaying the instruction over webcam input. While at first this may not seem a particularly useful deliverable, it represented a significant step in the right direction, as it allowed the current fold to be more easily compared with the instruction set at that step by putting the two images in the same place.



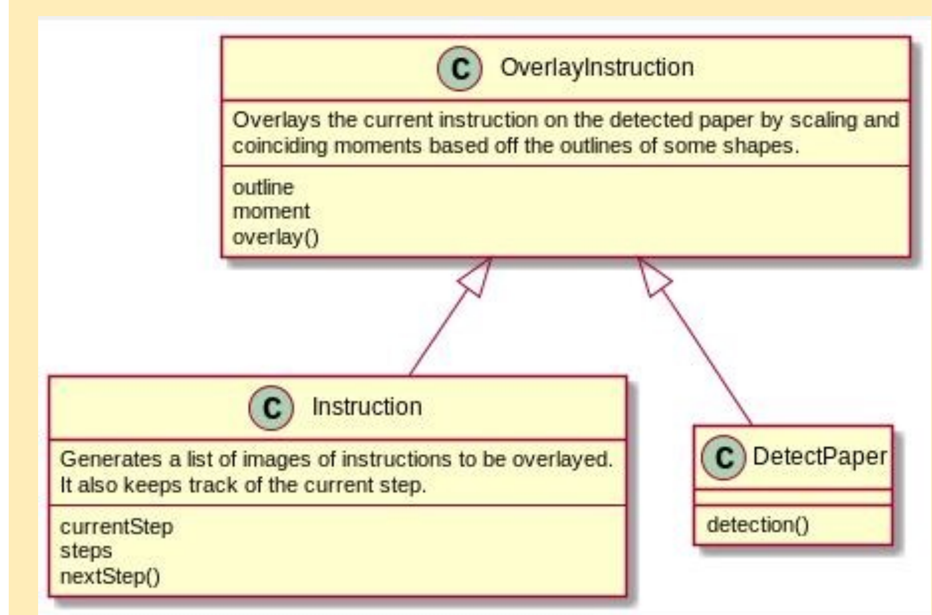
The simple method lies in first scaling the instruction image and then moving it to the correct location on the camera input. The camera input is then made into a black and white image in which the paper is made more apparent. Afterwards, the vertices are detected and the area of the shape is found using the shoelace formula. The same area-finding procedure is used on the image of the instruction image. By comparing the areas, the instruction image can be scaled to the size of the user's paper. The placement of the image of the instruction is done by overlaying the centroids of the vertices of the scaled instruction image and the user's paper.



The purpose of this third approach is to provide an augmented origami experience, where the instruction seems to become a part of the paper at every step, fading into what we believe could be a new way to experience origami altogether. Camera input is first processed into a black and white image that enhances the features of the paper relative to the background. From this image, contours that might be that paper are detected, and the largest contour with the same number of sides as the instruction shape is chosen. Finally, homography is used to project the instruction shape within the contour detected as the paper, such that it appears the paper itself is the instruction.

The Architecture

How it all came together:



Starting at the leaves of this tree, the Instruction class looks into a directory and finds all files with a name like step01.png. It then creates a list of these images in steps and keeps track of the current step number in currentstep. A program that uses this class uses nextStep to get the next instruction image. The following implementations are built atop this architecture, and provide varying levels of assistance and types of utility in the origami experience.