Computer Vision

Semester Project Part 4 Deliverable

Daniel McCarthy and Chris Hunt

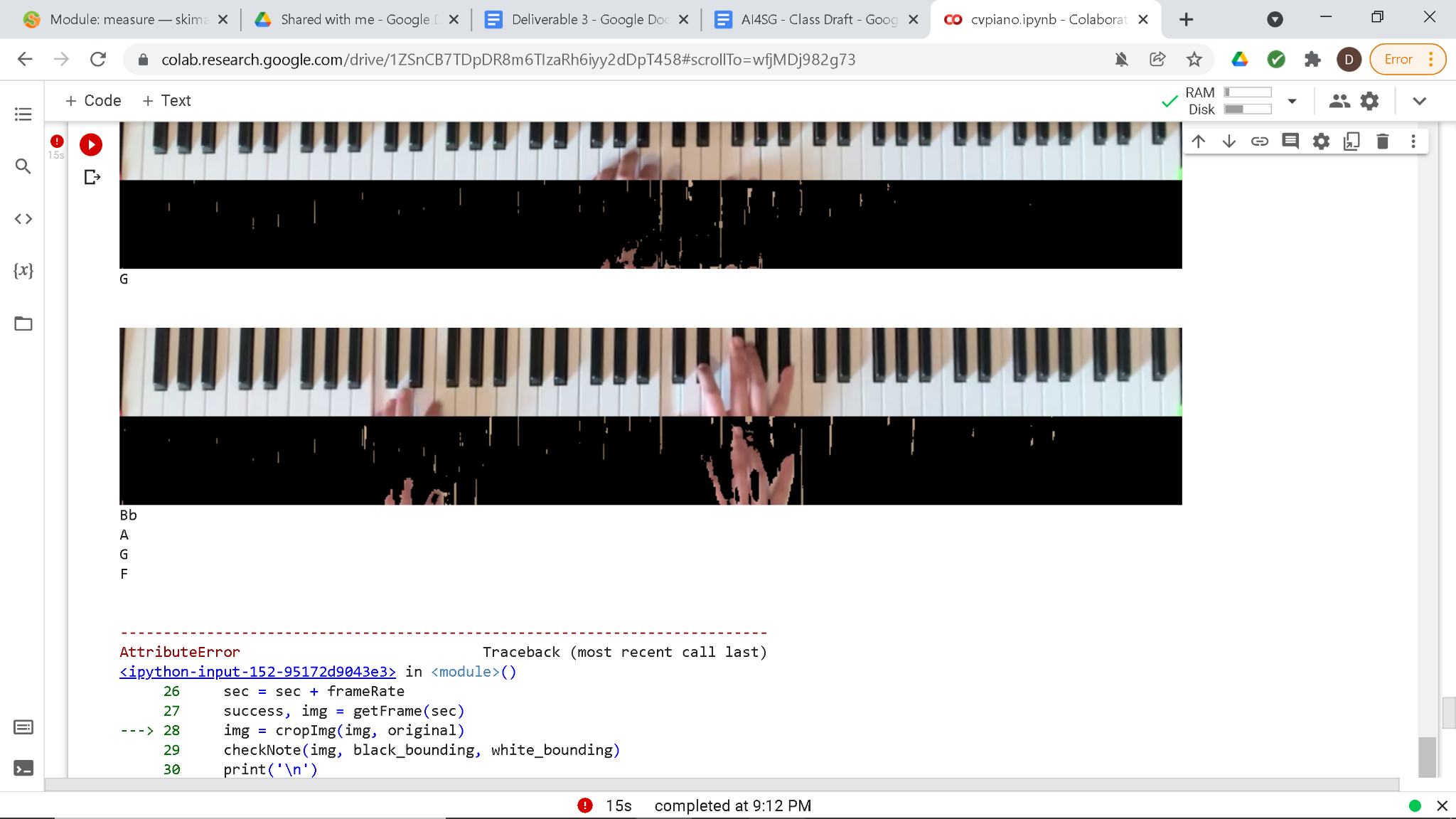
Final classification results

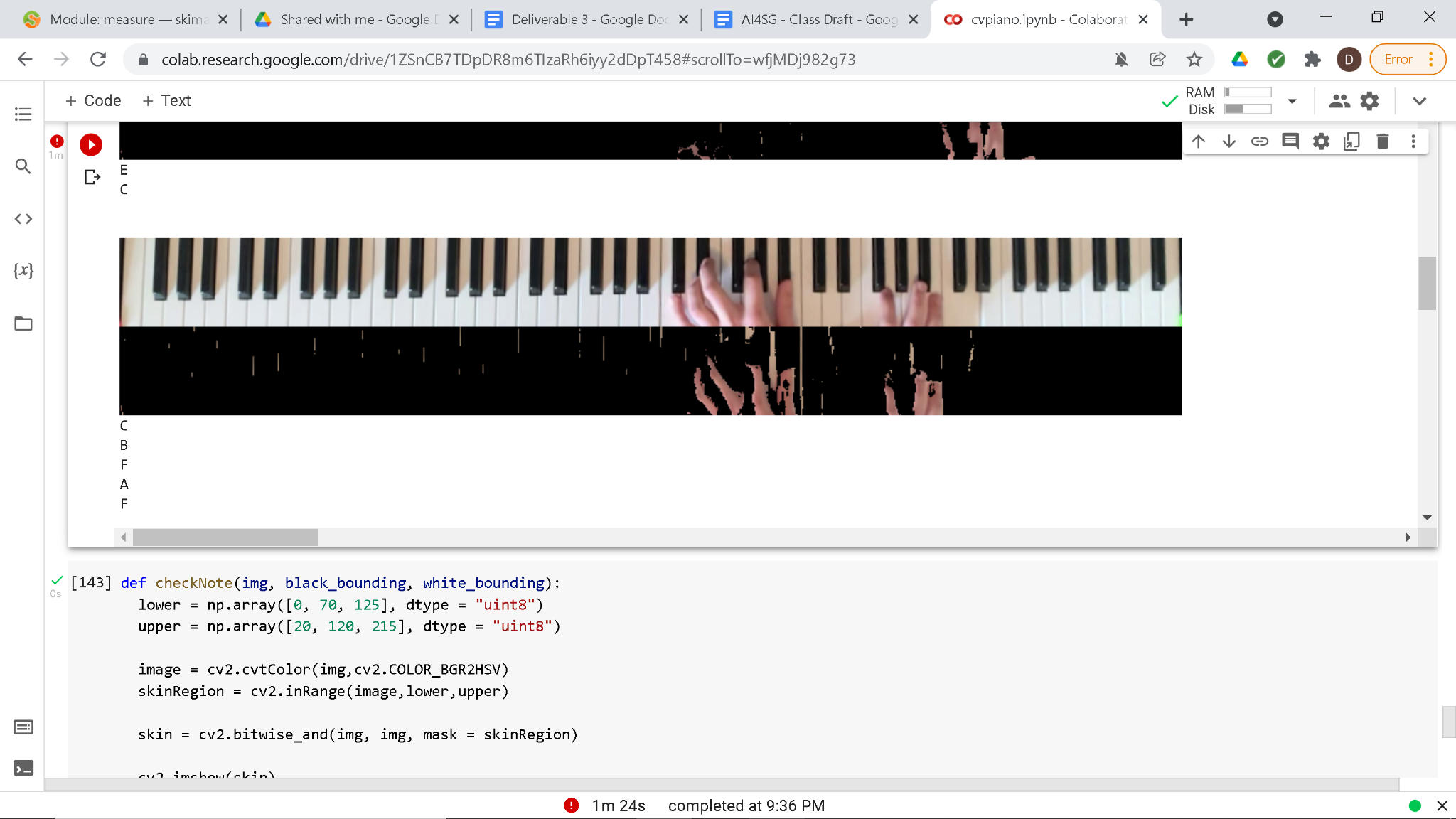
1. Method

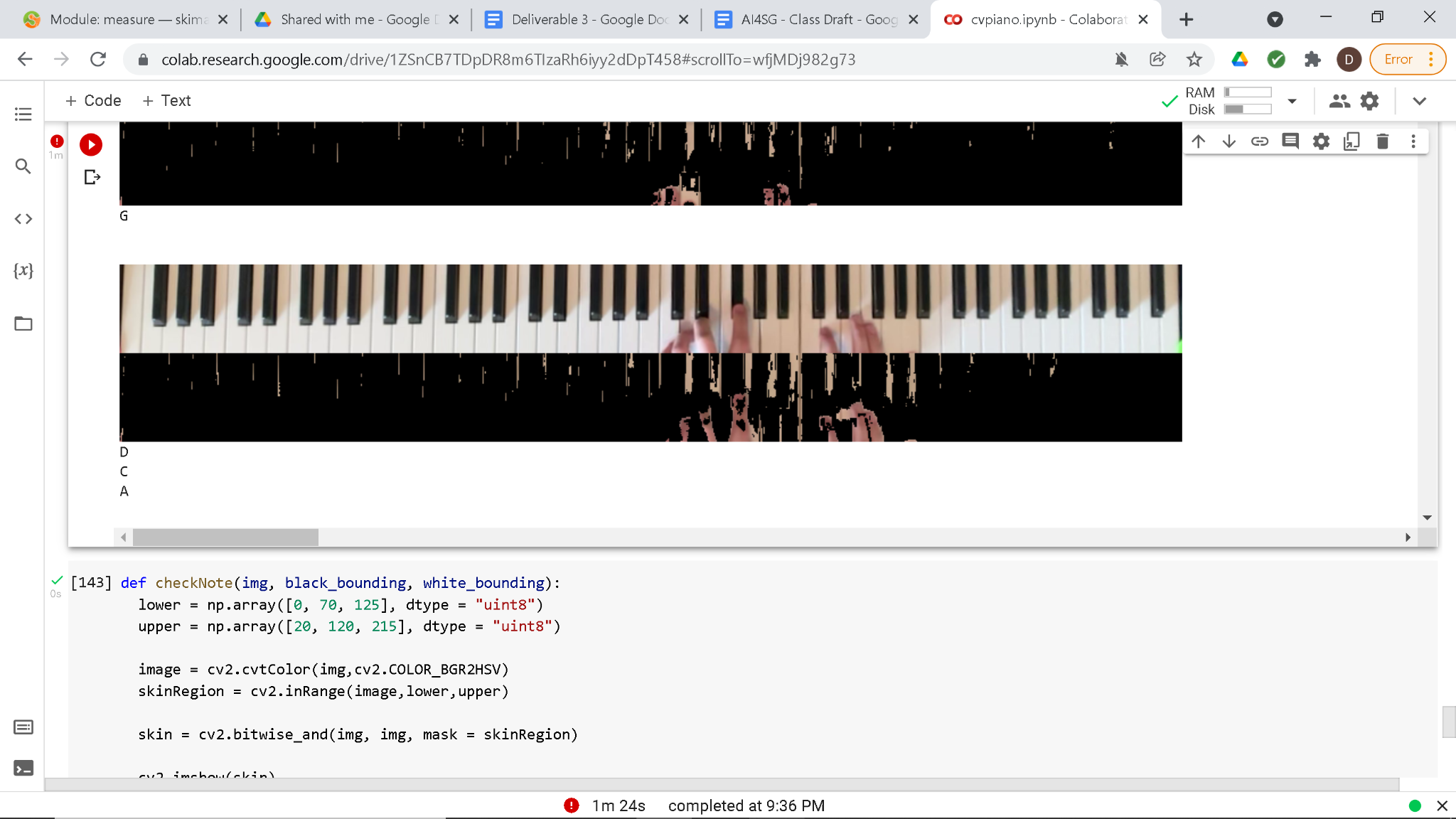
For our final project, we utilized Python computer vision libraries to create a functional piano key recognition system. The system operates by locating key points on the keyboard via colored stickers. Using these colored stickers, the keyboard image is straightened and cropped so that key detection can be done more easily. After cropping is complete, canny edge detection and hough line transforms are used to further identify where the top of the keys are and crop to create an image of only keys. Next, the cropped image is binarized and then an inverting mask is applied to isolate black keys from white. We find relative positions of white keys from the distances between the centroids of the black keys. Larger gaps between black keys are indicative of two white keys existing between them. Bounding boxes are finally drawn around the keys to be used for press detection.

Press detection has changed since our last deliverable. Following suggestions, we attempted to implement hand recognition using mediapipe’s library <https://google.github.io/mediapipe/solutions/hands.html>. However, we found that this implementation is only effective when the entire hand is in frame rather than just the fingers. Because of this, we instead utilized skeletonization on our skin region to make the skin region smaller and better identify where fingertips are located. After making the skin region smaller, we found that note detection actually suffered greatly (see improvements for more details). Our final method now remains the same as in deliverable 3 due to being unable to implement a better method despite our best efforts.

1. Results





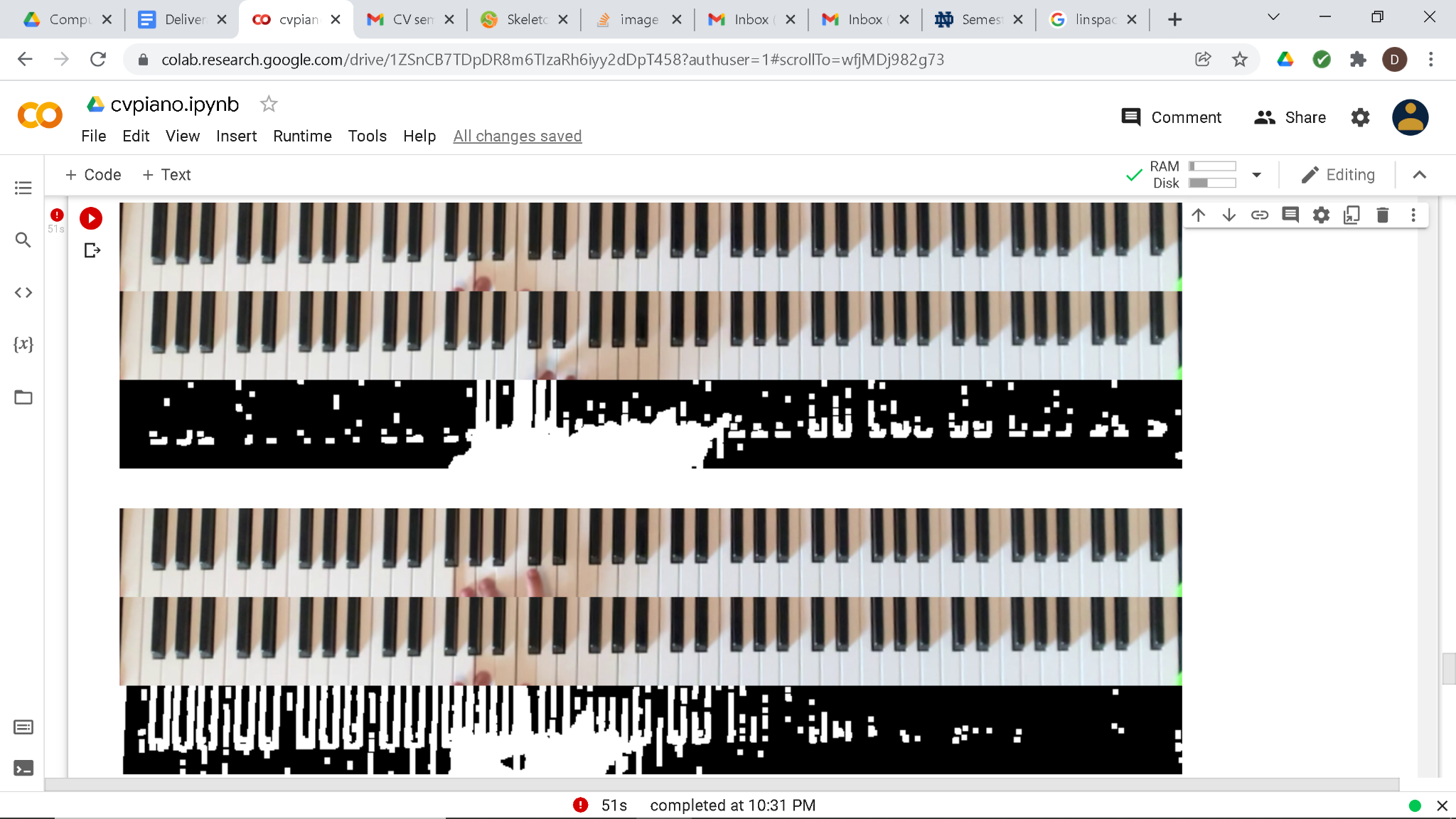


Since our note detection method is the same as deliverable 3, our results have remained the same as well. In the above images, you can see notes being played and the detected notes printed out beneath each image. The first image is the normal piano image with no alterations. The second is the normal image with the skin detection applied on top. The limitations of this method is apparent, some notes that are being played are not detected, and some notes that are not being played are detected. After filtering through our selection of played notes and visually checking to see if the correct detections appear or not, we estimate the accuracy of our program to be around 40%.

1. Improvements

Press detection was not implemented because of the finger problems and the general conditions that need to be satisfied in order to produce accurate results. With the lighting in the original video, many keys had a shadow on them to begin with, so the change in grayscale frames would already exist without even pressing keys yet. When keys were pressed, there were virtually no lighting changes on some keys that did not allow for proper detection. Since the pipeline should be general and not tailored to hyperspecific conditions, skin detection alone is the better option.

However, this means that in the future there would need to be something to account for false positives. A finger over the key does not automatically mean the player is pressing it.

Perhaps a keyboard that lights up when keys are pressed would make this feature work.

The below image displays skeletonization on our skin detected images. However, this skeletonization is not detected in our program when checking for which notes are being played, even though it is quite easy to see that there are skeletonized regions inside of the bounding boxes. Because of this difficulty, we have decided to maintain with skin detection as our note detection method.

