Full Implementation Code Documentation

Within this document, we list and describe the steps in the overall workflow used in the extraction of either driver licenses or vitals monitor. This implementation is a single application that takes in the zoom video recording from the smart glasses, and either outputs nothing (in the case of neither ID nor monitor being present), outputs the information from a driver's license (in the case of an ID being present), outputs the information from a vitals monitor (in the case of a monitor being present), or both (in the case of sequential presence). As an overview, there are four main steps in order to extract information from vitals or IDs:

1. **Detect the presence of either vitals or ID.** Each frame is fed into both individual YOLO algorithms found in the **Vitals Monitor Extraction** and **Drivers License Extraction** folders. Once an object is detected, we proceed to the next step.

2. **Continuously Run Only the Detected Algorithm Flow.** It is assumed that the object will be within frame for longer than one frame. Therefore, while the object of interest is being detected, in order to preserve processing speed, only the respective algorithm workflow will be run. For as many frames as the object is detected (with some lee-way for when the algorithm mistakenly does not detect the object for one or two frames), its algorithm flow will continuously be run, and each frame's OCR result will be obtained and recorded.

3. **Integrate Multiple Frames to Increase Accuracy.** In order to increase the accuracy of the program, reduce the effect of glare, and exploit the fact that multiple frames are used per object, the algorithm runs an integration algorithm that:
    1. Takes the string outputs of the OCRs of multiple frames with respect to a information category (ex. Birth Date, Heart Rate). For instance, we could look at each frame's Birth Date result.
    2. Creates weights for each frame's string output, depending on the focus/blurriness of the image of the string. For instance, we may have one blurry frame, one in-focus frame, and one in-between that each have a reported Birth Date. More weight would be given to the in-focus frame, some would be given to the one in-between, and the least weight would be given to the blurry frame.
    3. Using the weights and the string outputs, we estimate a running actual value of the information. This estimation also considers if some frames have read too many or too few characters.

    In the case of the ID, once the object is no longer detected for some time, the running values in step 3 will be reported. In the case of the vitals monitors, since vitals information will change, frames that were taken in the past will gradually lose weight, and the running estimation of the vitals will continuously be reported.

4. **Reset All After Object has Left the Frame.** Everything will be reset once the object has left the frame.

Overall Performance Table:

Video	Target Accuracy (or Loss)	Recorded Accuracy (or Loss)
Driver Licenses	95%	95%
Vitals Information	95%	99%

NOTE: Driver License Extraction and Monitor Vitals Extraction documentation can be found in their respective folders.

Frame Integration Algorithm

1. **Representation:** Every character, x, is represented as a vector of probabilities, each probability indicating a certain letter.

$$x = \left(x_1, x_2, ..., x_K\right) \in [0.0, 1.0]^K, \quad \sum_{k=1}^{K} x_k = 1,$$

   Every text, or string of characters, is therefore represented as a 2D matrix, where each character estimate is stacked on top of one another.

$$X = \left(x_{jk}\right) \in \left[0.0, 1.0\right]^{M \times K}, \quad \forall j : \sum_{k=1}^{K} x_{jk} = 1,$$

2. **Weights:** The weights of each frame are determined based on the focus estimation of an image, which is based on the MINIMUM of the UPPER 95% QUARTILE of the GRADIENTS of a string.

$$G^{V}_{\mathrm{r,c}}(I_i(\bar{X})) = \left|\mathrm{I}_{r+1,\mathrm{c}} - \mathrm{I}_{\mathrm{r,c}}\right|,$$
$$G^{H}_{\mathrm{r,c}}(I_i(\bar{X})) = \left|\mathrm{I}_{\mathrm{r,c}+1} - \mathrm{I}_{\mathrm{r,c}}\right|,$$
$$G^{D_1}_{\mathrm{r,c}}(I_i(\bar{X})) = (1/\sqrt{2})\left|\mathrm{I}_{r+1,\mathrm{c}+1} - \mathrm{I}_{\mathrm{r,c}}\right|,$$
$$G^{D_2}_{\mathrm{r,c}}(I_i(\bar{X})) = (1/\sqrt{2})\left|\mathrm{I}_{\mathrm{r,c}+1} - \mathrm{I}_{r+1,\mathrm{c}}\right|,$$

Image Gradient (G) Calculation in the x,y,xy,yx direction, using image intensity (I).

$$F(I_i(\bar{X})) = \min\left\{q(G^{V}(I_i(\bar{X}))), q(G^{H}(I_i(\bar{X}))), \right.$$
$$\left. q(G^{D_1}(I_i(\bar{X}))), q(G^{D_2}(I_i(\bar{X})))\right\},$$

Focus Calculation using the minimum of the upper 95% quartile (q) of the gradients.

3. **Combination Process:** We begin with an empty column vector or the first recognized result. When two recognized texts are obtained, due to variable length recognition, the texts are first aligned using the distance equation below. In essence, the algorithm aligns the text to maximize agreeability.

$$\rho\left(x^1, x^2\right) = \frac{1}{2}\sum_{k=0}^{K}\left|x_k^1 - x_k^2\right|$$

Then, the algorithm combines the two texts using the weighted equation below to produce a resulting estimation of the desired text.

$$r = \left(r_k\right) \in \left[0.0, 1.0\right]^{K+1}, \quad \forall k : r_k = \frac{x_k^1 \cdot w(x^1) + x_k^2 \cdot w(x^2)}{w(x^1) + w(x^2)},$$

This process is repeated, with the running estimation being the ‘first frame’, and new frames being the ‘second frame.’

Testing Protocol

1. Two videos were taken with the smart glasses and recorded using zoom in order to assess the performance of the algorithm.
    a. The first video (**Video/driver.mp4)** begins in an empty environment. Then, sequentially, driver licenses made of paper are brought into frame for ~3-5 seconds and then taken out of frame for ~3-5 seconds. Reports of the driver

licenses are output in a text file. Accuracy is determined by correct results/all results.

   i. Accuracy = 95%

b. The second video (**Video/vital.mp4)** begins by observing a vitals monitor. Throughout the video, the camera moves around this monitor, during which the vitals information changes. Accuracy is determined based on the reported vitals information across time.

   i. Accuracy = 99%