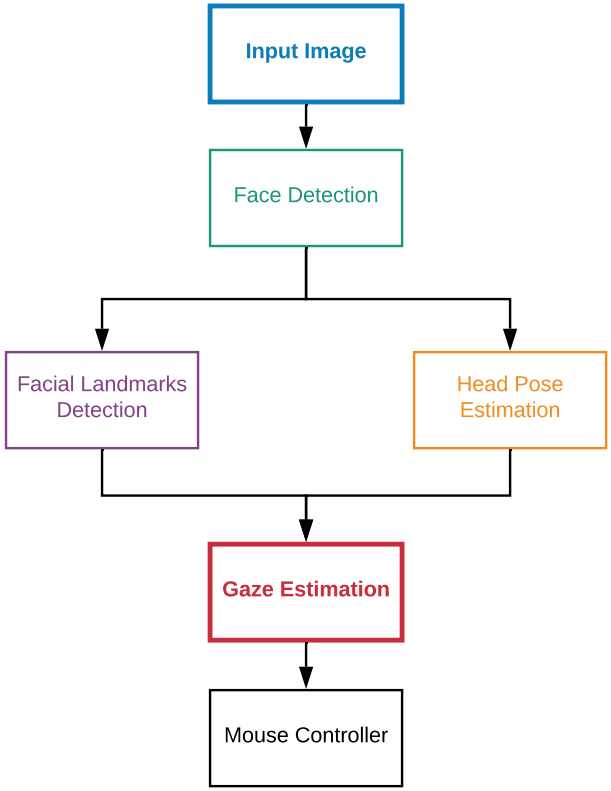
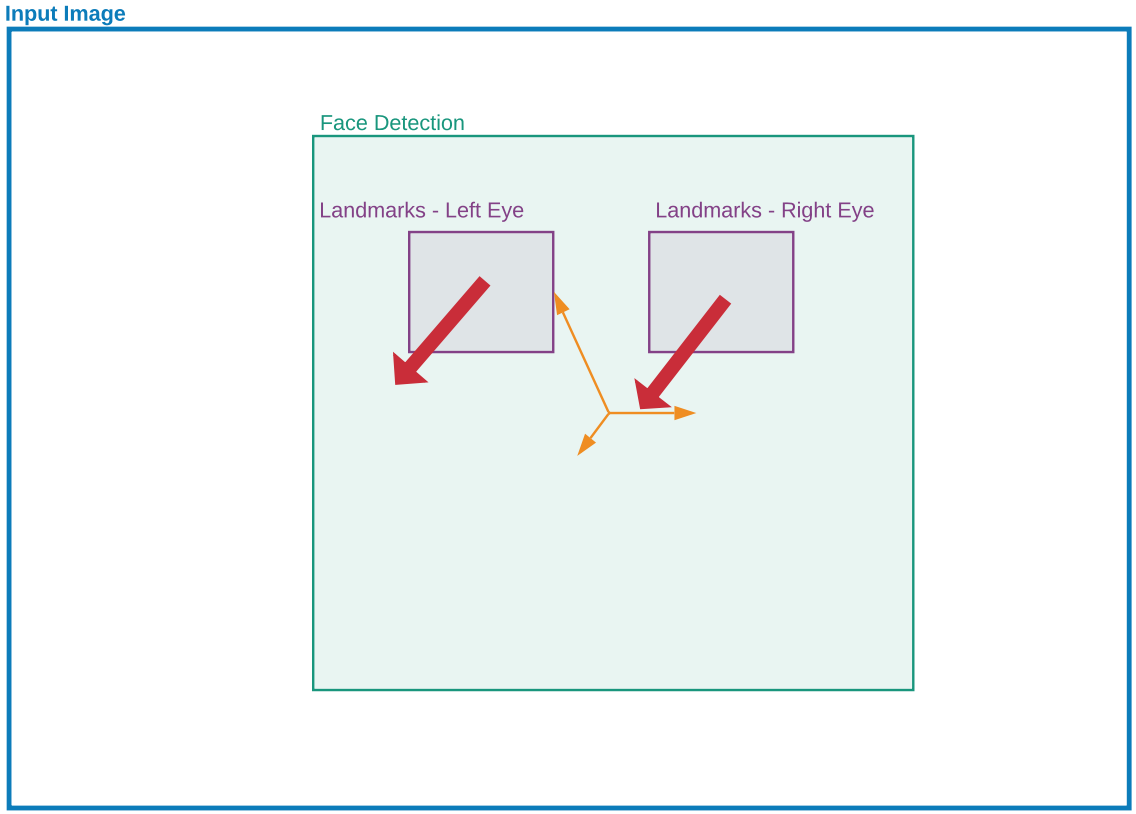
Computer Pointer Controller – Less Formal Overview

Cedric Membrez | June 14, 2020





	INPUTS	OUTPUTS
FaceDetection.predict(**INPUTS**) return **OUTPUTS**	**from:** InputFeeder.next_batch() **format:** a frame **shape:** (InputFeeder's width, InputFeeders's height, # channel)	**Coordinates:** [(x0, y0), (x1, y1)]; if no detection *None* **Cropped Prediction:** batch is cropped on prediction where top-left is (x0, y0) and bottom-right is (x1, y1); if no detection, return original batch
FacialLandmarks.predict(**INPUTS**) return **OUTPUTS**	**from:** FaceDetection.predict()'s outputs, and original batch **format:** a cropped frame of Face, and original batch **shape:** (face's width, face's height, #channel), (original batch's width, & height, #channel)	**Coordinates:** [(x0, y0), (x1, y1)] **Cropped Eyes:** one cropped frame around each eyes, i.e. point_left_eye=(x0, y0) and point_right_eye=(x1, y1) coordinates are in range [0,1] and based on cropped_face_frame --> (coord_x * cropped_x) + face_x = coord_normal_x where cropped_x := side (length) x of the cropped face frame, and face_x is the coordinate x of the detected face.
HeadPoseEstimation.predict(**INPUTS**) return **OUTPUTS**	**from**: FaceDetection.predict()'s outputs **format**: a cropped frame of Face **shape**: (face's width, face's height, #channel)	**Head Angles:** return Tait-Bryan angles in degree in the form (z, y, x), i.e. (yaw, pitch, roll)
GazeEstimation.predict(**INPUTS**) return **OUTPUTS**	**from:** FacialLandmarks.predict()'s outputs, HeadPoseEstimation.predict()'s outputs **format:** (left eye cropped frame, right eye cropped frame), (head angles array) **shape:** [(60x60x3), (60x60x3)], [z, y, x]	**Gaze Coordinates:** a 3-D coordinate (x, y, z) of the user's gaze. 1) get coordinates of eyes and make them 'global' 2) compute the middle point between the eyes 3) from this (x_mid, y_mid), add the (x_gaze, y_gaze) as offset. 4) draw the arrow