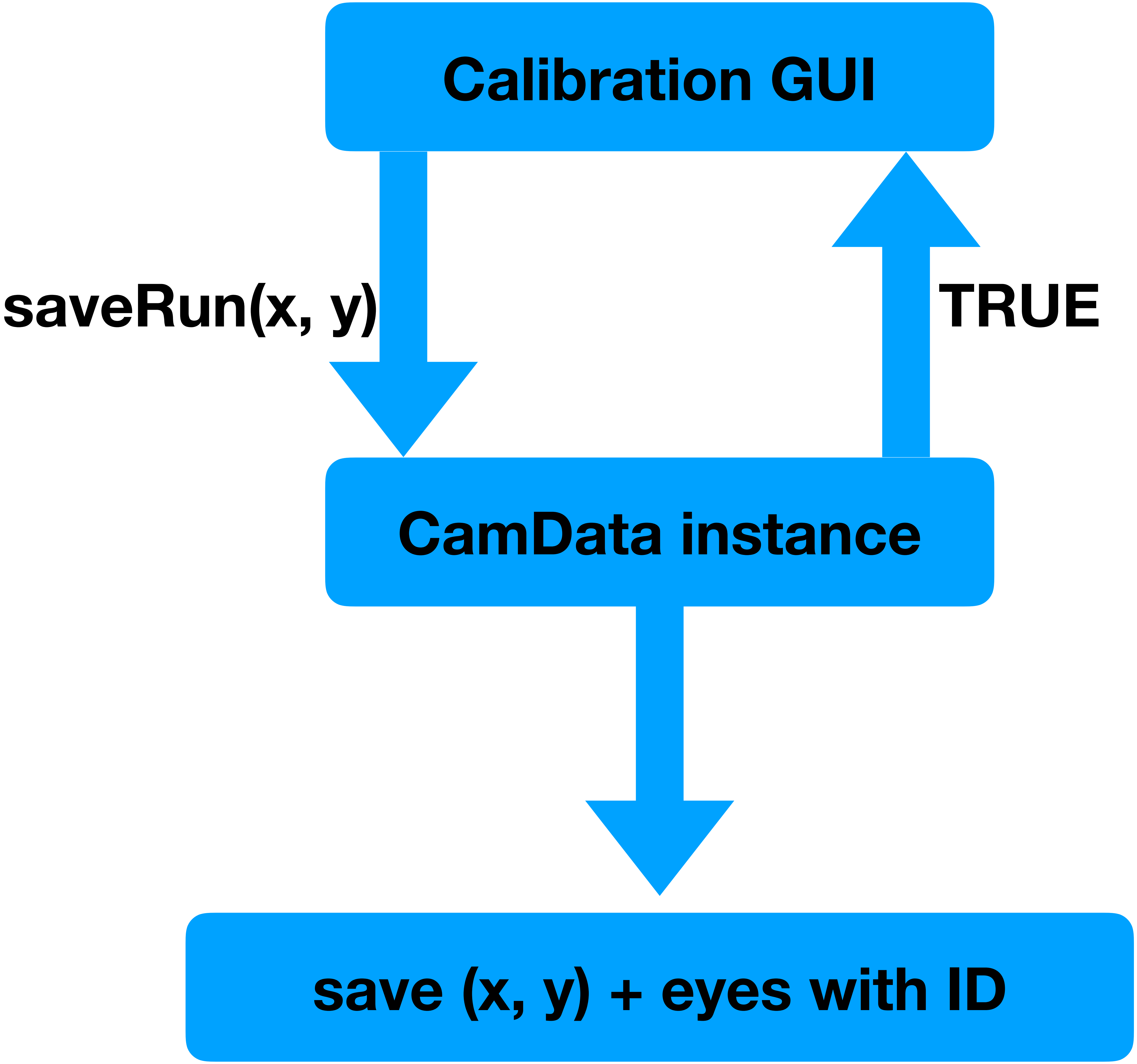


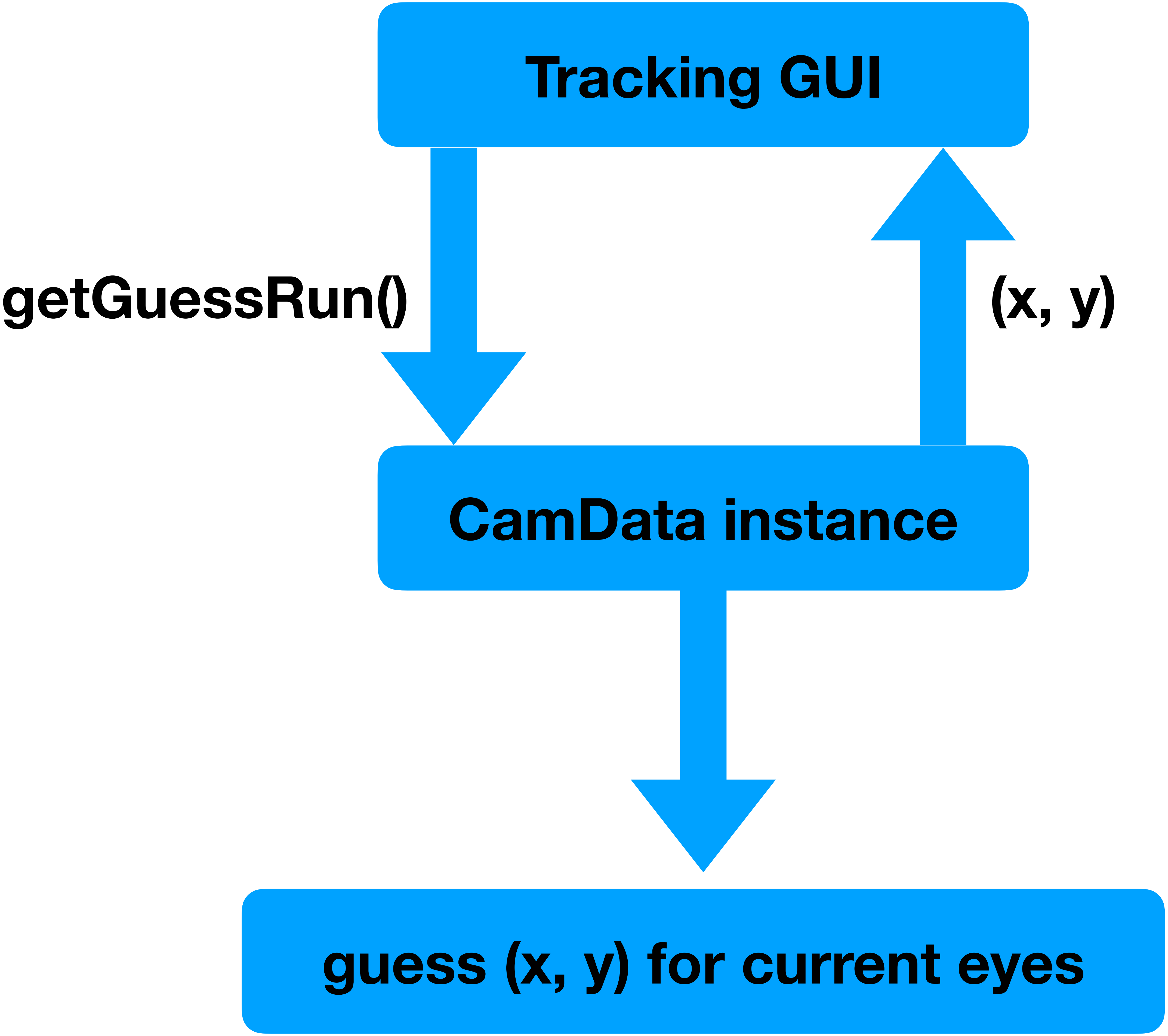
Calibration GUI

saveRun(x, y)

TRUE

CamData instance

save (x, y) + eyes with ID



Tracking GUI

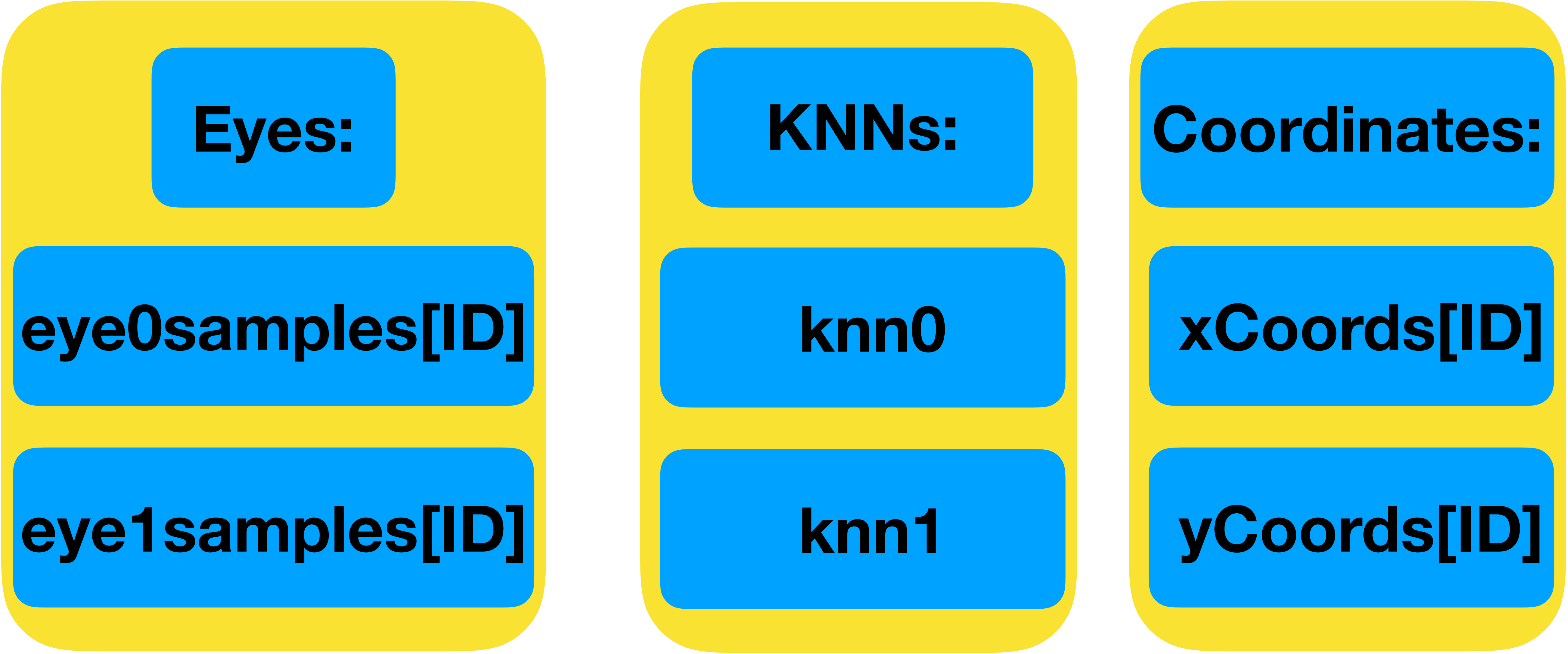
getGuessRun()

(x, y)

CamData instance

guess (x, y) for current eyes

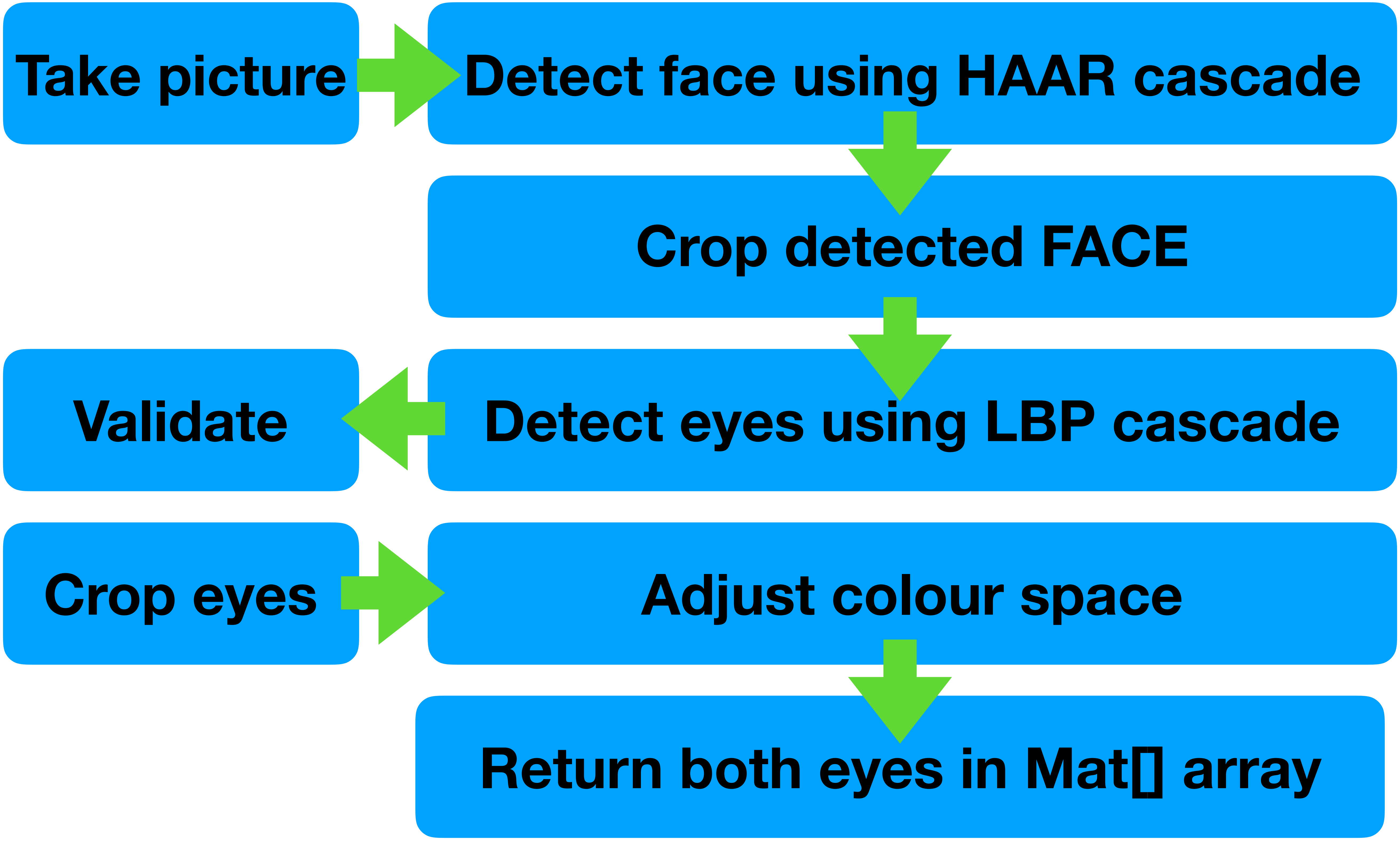
CamData instance calibration storage



Eyes:	KNNs:	Coordinates:
eye0samples[ID]	knn0	xCoords[ID]
eye1samples[ID]	knn1	yCoords[ID]

Everything accessible by ID

CamData .getEyesImprovedRun() method:



Take picture

Detect face using HAAR cascade

Crop detected FACE

Detect eyes using LBP cascade

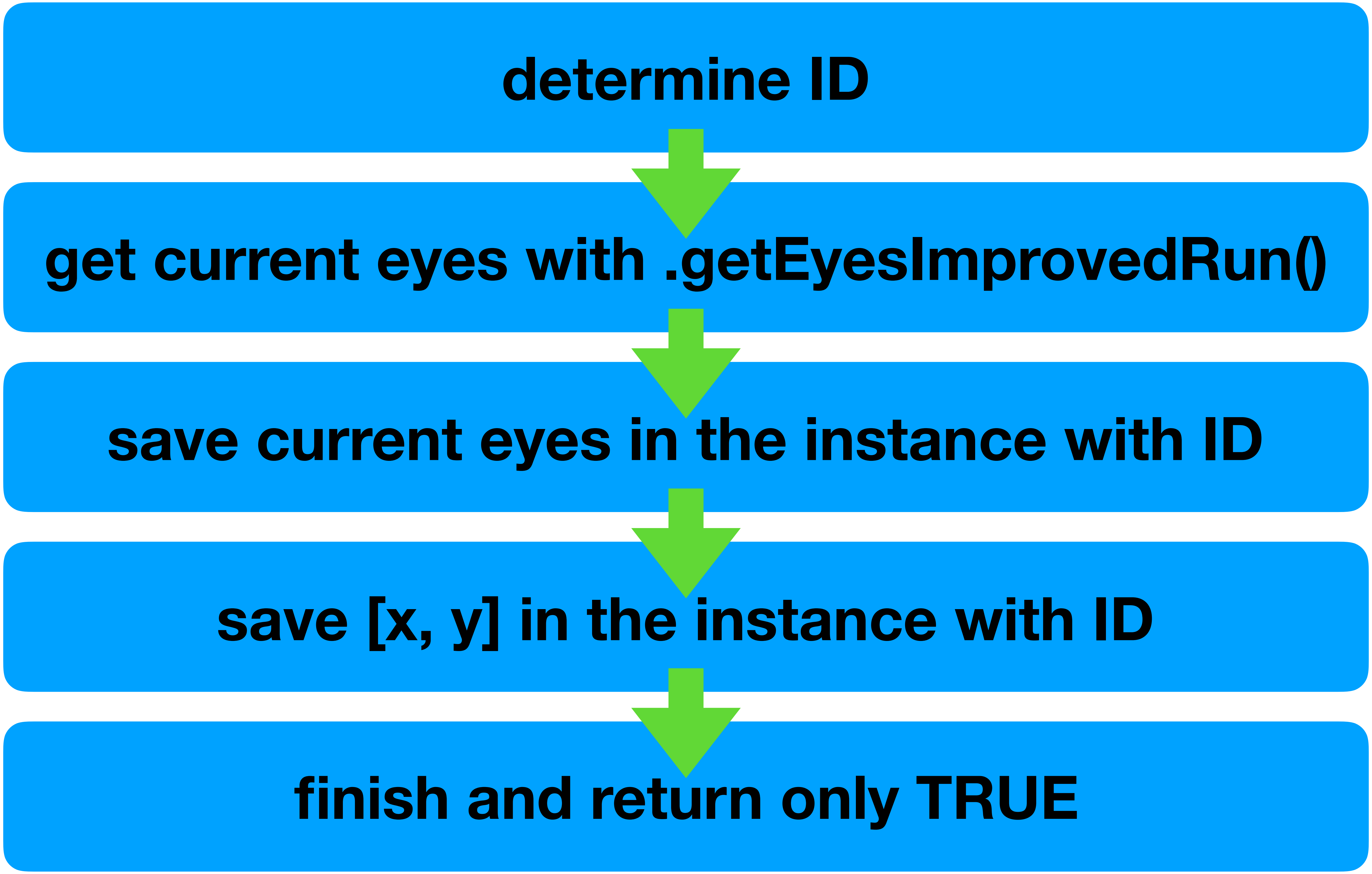
Validate

Crop eyes

Adjust colour space

Return both eyes in Mat[] array

CamData .saveRun([x, y]) method:



determine ID

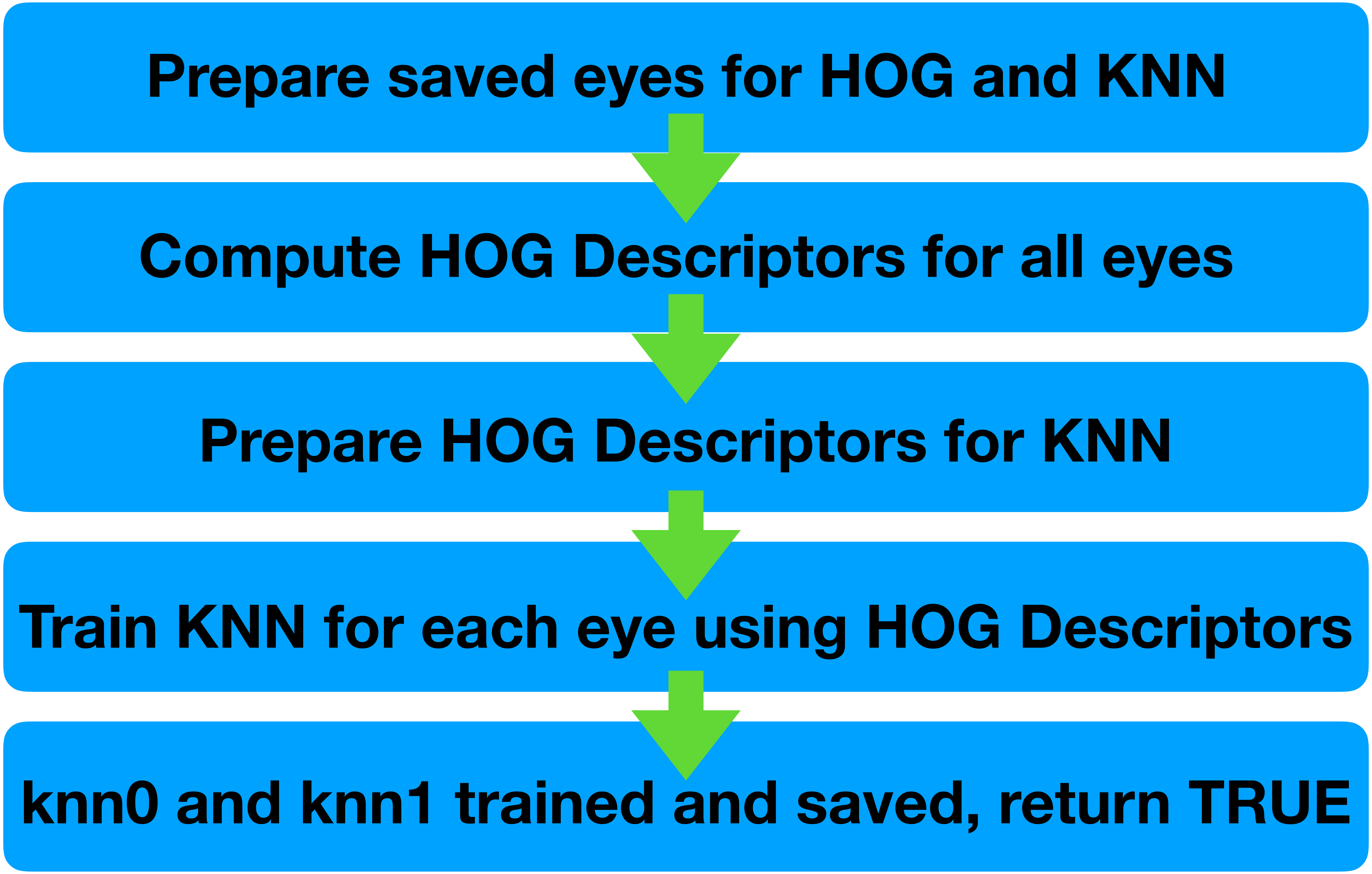
get current eyes with .getEyesImprovedRun()

save current eyes in the instance with ID

save [x, y] in the instance with ID

finish and return only TRUE

CamData .initializeKNearest() method:



Prepare saved eyes for HOG and KNN

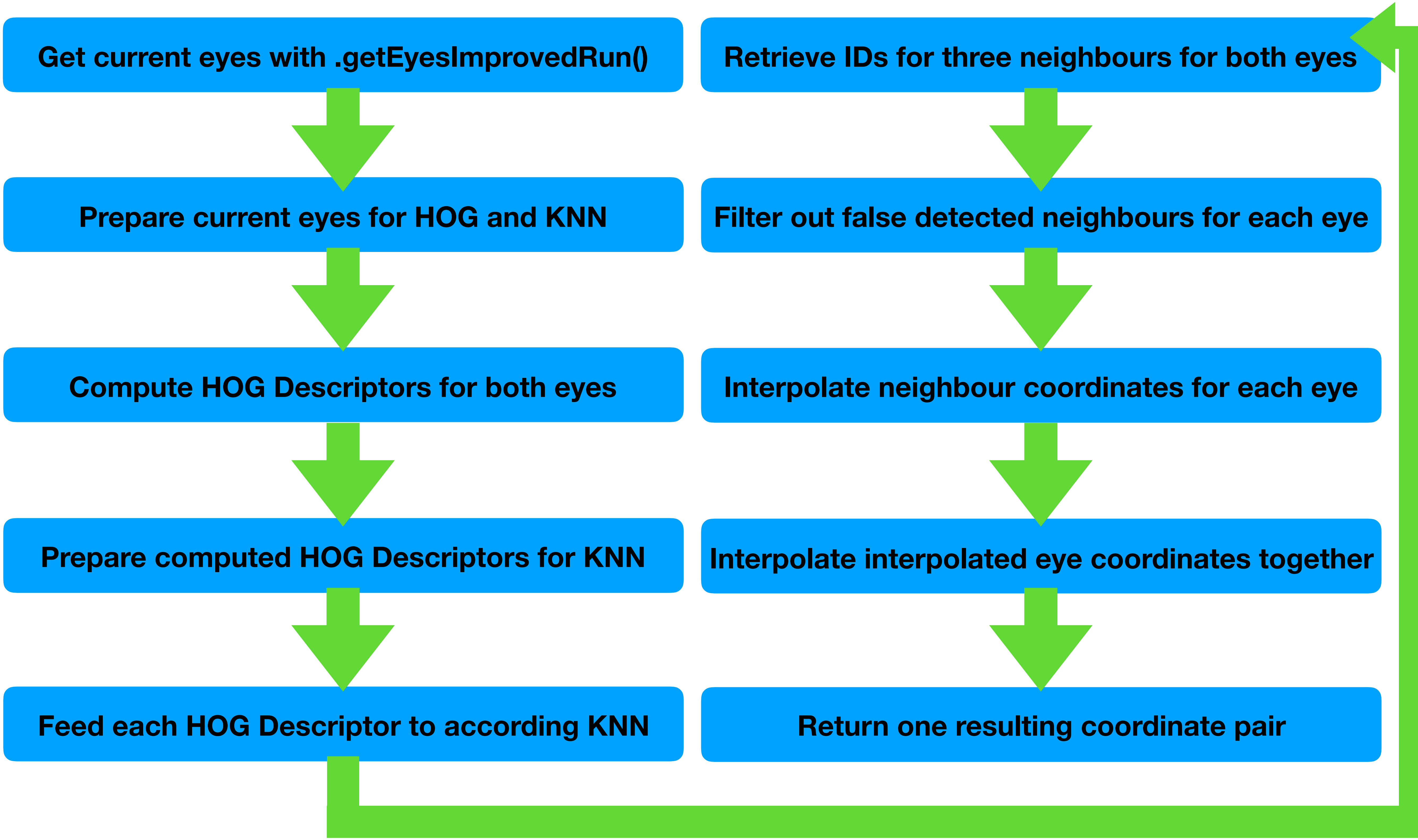
Compute HOG Descriptors for all eyes

Prepare HOG Descriptors for KNN

Train KNN for each eye using HOG Descriptors

knn0 and knn1 trained and saved, return TRUE

CamData .getGuessRun() method:



Get current eyes with .getEyesImprovedRun()

Prepare current eyes for HOG and KNN

Compute HOG Descriptors for both eyes

Prepare computed HOG Descriptors for KNN

Feed each HOG Descriptor to according KNN

Retrieve IDs for three neighbours for both eyes

Filter out false detected neighbours for each eye

Interpolate neighbour coordinates for each eye

Interpolate interpolated eye coordinates together

Return one resulting coordinate pair