UML Project Class Diagram

class VideoCapture

class CascadeClassifier

class shape predictor

2

Class FaceDetector

- CamFrame : std::unique_ptr<cv::VideoCapture>
- Face_Cascade : std::unique_ptr<cv::CascadeClassifier>

- AdjustedFrame : cv::Mat
- FixedWidth = 256 : const int
- FaceRects : std::vector<cv::Rect>
- TempFacesRect : std::vector<cv::Rect>

- INRoi : std::vector<bool>
- FaceTemp : std::vector<cv::Mat>
- FaceRegionOfInterest : std::vector<cv::Rect>
- MatchingResult : cv::Mat
- AdjustedSize : cv::Size
- OriginalFrameSize : cv::Size
- RatioForAdjustment : cv::Point2f

- Istracking = false : bool
- FaceFound = 0 : unsigned int

- BeginAddface : std::vector<long long>
- StopAddFace : std::vector<long long>
- MaxTimeOnFrame = 2.0 : const double

+ FaceDetector(const std::string cascadeFilePath, const int cameraIndex, size_t numFaces);
+ ~FaceDetector();
+ operator >> (cv::Mat &frame) : void
+ faces() : std:: vector<cv::Rect>
+ doubleRectSize(const cv::Rect &rect, const cv::Size &frameSize) : static cv::Rect
+ detect() : void
+ track() : void
+ instruction() : virtual void

Class SaveFrame

- filenum : int
- imagequality : vector<int>
- Frame : Mat
- folderpath : string
- filename : string

+ SaveImage(Mat& frame,string imagename);
+ SaveImage(string Path,string imagename, Mat& frame);

+ setFolderPath(string Path) : void
+ printFilename() : void

+ getFolderPath() : string
+ getFrame() : Mat

+ getImageQuality() : vector<int>
+ setImageName(string imagename) : void
+ SaveImagetoFile(Mat Frame) : void

+ intToString(int number) : string
+ setSaveFile(int fnum) : void

Class ColorMapping

- colorframe : int
- inputFrame : Mat
- OutputFrmae : Mat
- colorlist : ListColor
+ enum ListColor

+ ColorMapping(ListColor cmap);
+ ConvertColor(Mat inputframe,Mat outputFrame) : void
+ setColorTone(ListColor color) : void

Class FaceSwapper

- face1 : cv::Rect
- face2 : cv:: Rect
- ReshapedFace1 : cv::Rect
- ReshapedFace2 : cv::Rect
- face_model : dlib::shape_predictor
- ShapeDetection[2] : dlib::full_object_detection
- dlib_rects[2] : dlib::rectangle
- dlib_frame : dlib::cv_image<dlib::bgr_pixel>
- affine_transform_keypoints_Face1[3] : cv::Point2f
- affine_transform_keypoints_Face2[3] : cv :: Point2f
- Alpha_Reshaped_face1_to2 : cv::Mat
- Alpha_Reshaped_face2_to1 : cv::Mat
- warpped_Face1, warpped_Face2 : cv::Mat
- Points_on_Face1[9], Points_on_Face2[9] : cv::Point2i
- Add_Face1_to_Face2 : cv::Mat
- Add_Face2_to_Face1 : cv::Mat
- Face1_Mask : cv::Mat
- Face2_Mask : cv::Mat
- ReshapedMask_face1 : cv::Mat
- ReshapedMask_face2 : cv::Mat
- RealFace1 : cv::Mat
- RealFace2 : cv::Mat
- AllReshaped_face : cv::Mat
- AdjustedFrame : cv::Mat
- FrameSize : cv::Size
- feather_amount : cv::Size
- LookUPTable[3][256] : uint8_t
- source_hist_int[3][256] : int
- target_hist_int[3][256] : int
- source_histogram[3][256] : float
- target_histogram[3][256] : float

+ FaceSwapper(const std::string landmarkFile);
+ ~FaceSwapper();
+ swapFaces(cv::Mat &frame, cv::Rect &face1, cv::Rect &face2) : void
+ getMinFrame(const cv::Mat &frame, cv::Rect &rect_face1, cv::Rect &rect_face2) : cv::Mat
+ getFacePosition(const cv::Mat &frame)   : void
+ getTransformationMatrices() : void
+ getMasks() : void
+ getReshapedMasks() : void
+ getAdjusted_ReshapedMasks() cv::Mat
+ extractFaces()   : void
+ getReshapedFaces() : cv::Mat
+ colorCorrectFaces() : void
+ featherMask(cv::Mat &refined_masks) : void
+ pasteFacesOnFrame()   : void
+ CalculateHistogram(const cv::Mat source_image, cv::Mat target_image, cv::Mat mask) : void

---

Class FaceFilter

- MAX_FACE_SIZE = 200 : double
- MIN_FACE_SIZE = 20 : double
- facefile : string
- faces : vector<Rect>
- Mask : Mat

- ResizedMask : Mat
- TempSourceFrame : Mat
- TransparentWhite : Mat
- TransparentedMask : Mat
- MaskAlpha : Mat

+ FaceFilter(const string facefile);
+ ~FaceFilter();

+ getFaceFile() : string
+ setMask(string f) : void
+ ReadMaskFile() : void

+ LocateTheMask(Mat frame) : cv::Mat
+ CreateMask(Mat& source_frame, Point face_centre, Size face_size) : cv::Mat