

AMENDED Exhibit 879

PLAINTIFFS' OMNIBUS OPPOSITION TO DEFENDANTS' MOTIONS FOR SUMMARY JUDGMENT

Case No.: 4:22-md-03047-YGR

MDL No. 3047

In Re: Social Media Adolescent Addiction/Personal Injury Products Liability Litigation

Face Analysis Library V1

8/24/2015, 5:43 PM UTC

Labels

type: document

Quip

Face Analysis Library V1

This spec intends to document the functionalities, usages and its applications .

Feature Goals

- Used on **Client Only**
- Detect faces snaps or story videos
- Detect facial landmarks (left corner of left eye, right corner of left eye, ..., right corner of mouth, etc), currently detected 68 facial landmarks
- Aligned face
- Face attributes prediction:
 - Facial landmark detection
 - Face frontalization
 - Gender prediction
 - FaceType Prediction
 - Age prediction
 - Face feature extraction
 - Emotion recognition (currently only happiness)

Owners

- Research: [REDACTED]

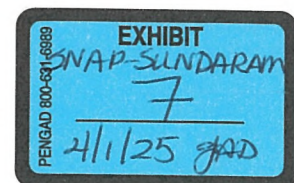
Terms

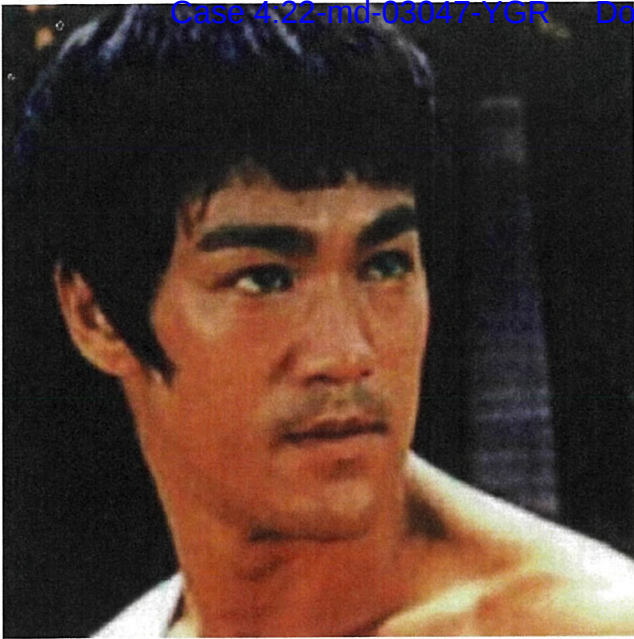
- **Bounding box**: The tight region around the face
- **Facial landmarks**: The most distinctive corners on the face
- **prob**: Probability

Resources

Functions

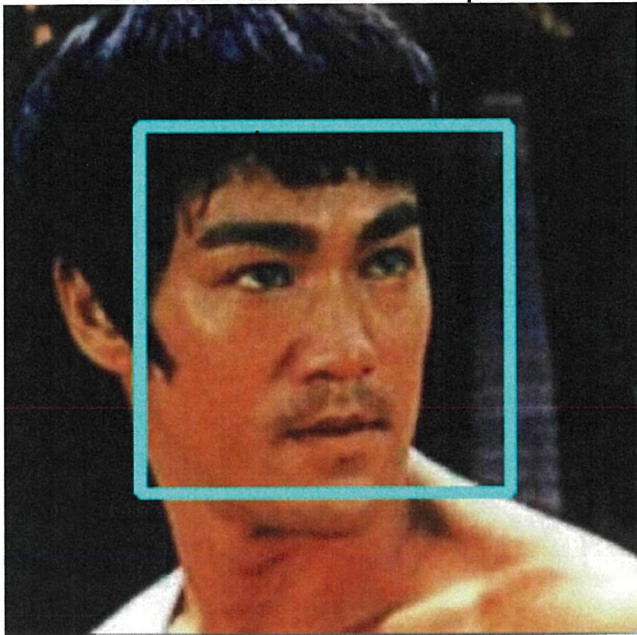
The following image will be used as an example to show the results of different functions.



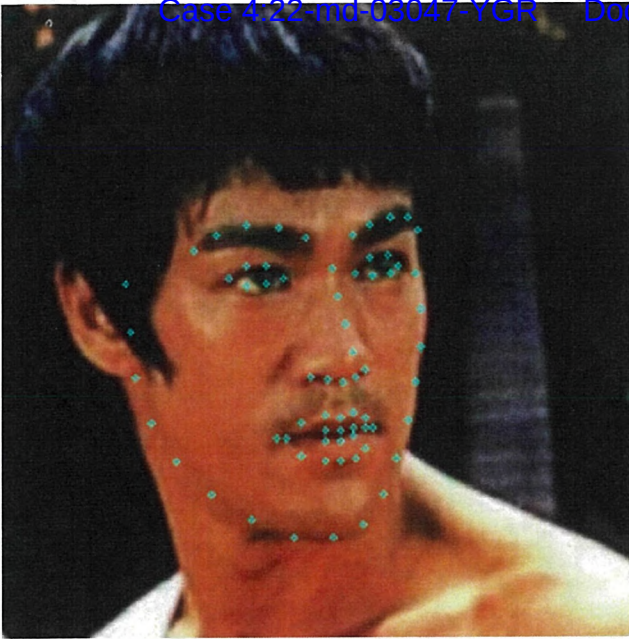


Currently, the Face Analysis Library provides the following functions:

- Face detection: Detect multiple faces in a snap or frames of videos



- Facial landmark detection: Generate 68 landmarks on the face as in the following example.



- Face alignment
 - Method: Using the facial landmark information to warp and crop the face.



- Attributes Prediction
 - Gender: male(prob: 0.94)
 - Age: 40
 - FaceType: [1.0, 0.0, 0.0, 0.0]
 - Emotion Recognition: (Happiness prob: 0.01)
 - Method: A classification model is learned end-to-end to do the prediction from input aligned image.
- Face feature extraction
 - Face feature vector is a fixed length vector of numbers, it has the property of measuring how similar a pair of input image is. For example, if the similarity between feature vectors of two images are small, they are less similar, and vice versa.
 - Method: The aligned face image is passed through a Convolutional Neural Network, the output from a specified layer is extracted and used as a feature vector to describe the input face image.

Privacy Review

Redacted--AC

Redacted--AC

Once this section is completed and the spec is ready for review....

- Email [REDACTED] with...

Subject: (Name of this spec) Body: (Target Release and Link to Quip).

- Fill in review request details below:

Sheet1_2

	Review Requester	Date	Notes
1	[REDACTED]		

Review Results (law / sec)

Only **Security and Legal** should fill in this section. If Approved, any future changes will require a new review.

Sheet3

	Review Type	Reviewer	Date	Status	Notes
1	Privacy - Engineering	[REDACTED]	3/6/2017	Under Review	Spec needs to be updated
2	Privacy - Legal				

Applications

This section tracks known applications of this library.

- Snapmoji: [Snapmoji Avatar Generation Technology Spec \[v1\]](#)

Updates

All changes to the spec are tracked here. Tag interested parties to notify. Notified should ACK the change in LGTM.

NOTE: New content should always be inserted at the top

	Change description	Author	Date	People to notify	LGTM
1	Created Spec	[REDACTED]	6/15/2016		

Metadata of Face Analysis Library V1

Thread class

document

Created

2015-08-24T17:43:20+00:00

Updated

2021-06-27T02:17:46+00:00

Link

<https://snapchat.quip.com/bFtUAVipFFTD>

Shared folders

—

Expanded users

[REDACTED]

Users

[REDACTED]

Urls

- <https://quip.com/EdQAEArOZdO>
- <https://quip.com/aYFAEAm3Qyy>
- <https://quip.com/TOXAEAAcS6G>
- <https://quip.com/acZAEAoJNsj>
- <https://docs.google.com/spreadsheets/d/110nB4WZRLgBZDgbqKUhx2jYSIGEGC6jrLb6OZF7cR0c/edit#gid=0>
- [mailto:\[REDACTED\]](mailto:[REDACTED])
- <https://quip.com/NdfAEANEIYR>
- <https://quip.com/FRKAEAVwP7h>
- <https://snapchat.quip.com/m5BaAj7mjcp7>
- <https://quip.com/EdQAEArOZdO>

Chat

- [REDACTED]

[REDACTED] Do we have all these libraries in production quality? Thanks!

6/14/2016, 9:42 PM UTC

- [REDACTED]

No, let me remove some and only keep the product lml functions first. And then add more when they are getting product ready

6/14/2016, 9:47 PM UTC

- [REDACTED]

• Yep, that's my impression and let's keep the official quip as the production quality ones or those we are developing the production quality in the near term. Thanks!

6/14/2016, 9:55 PM UTC

- [REDACTED]

OK, sounds good. Already updated

6/14/2016, 9:55 PM UTC

- [REDACTED]

Redacted--AC

6/17/2016, 5:18 PM UTC

- [REDACTED]

Redacted--AC

Redacted--AC

6/17/2016, 5:31 PM UTC

- [REDACTED]

Redacted--AC

- [REDACTED]

Redacted--AC

6/28/2016, 5:00 AM UTC

- [REDACTED]

[REDACTED] Could you explain how you do age prediction here? (in technical terms).

7/12/2017, 8:50 PM UTC

- [REDACTED]

We are using a neural network model (NN) behind the scene. The NN takes an face image as input, and directly output the predicted age

7/12/2017, 8:53 PM UTC

- [REDACTED]

[REDACTED] Got it. Thanks!!

7/12/2017, 8:54 PM UTC

Redacted--AC

7/12/2017, 9:01 PM UTC

• [REDACTED]

Redacted--AC

7/12/2017, 9:01 PM UTC

Edits

[REDACTED] edited at 2017-07-12T21:01:49.762Z

Redacted--AC

[REDACTED] edited at 2017-07-12T20:49:52.612Z

[REDACTED] edited at 2017-07-05T16:41:15.880Z

- Design:
- Research: [REDACTED]
- QA:
- PM: [REDACTED]

- User Stories
- Face alignment
- Method: Using the facial landmark information to warp and crop the face.
- *Image*
- Method: A classification model is learned end-to-end to do the prediction from input aligned image.
- Face feature extraction
- Face feature vector is a fixed length vector of numbers, it has the property of measuring how similar a pair of input image is. For example, if the similarity between feature vectors of two images are small, they are less similar, and vice versa.
- Method: The aligned face image is passed through a Convolutional Neutral Network, the output from a specified layer is extracted and used as a feature vector to describe the input face image.

Xutao Lv edited at 2017-06-30T21:18:17.411Z

- Research: [REDACTED]
- QA:
- PM: [REDACTED]

edited at 2017-03-06T23:15:21.219Z

- Sheet3 updated

edited at 2016-07-21T22:48:57.766Z

- Gender prediction
- Race FaceType Prediction
- Age prediction
- Face feature extraction
- Gender: male(prob: 0.94)
- Age: 40
- Race FaceType: Asian(prob: 1 [1.0] 0, White(prob:0) 0.0, Black(prob: 0) 0.0, Indian(prob:0) 0.0]
- Emotion Recognition: (Happiness prob: 0.01)

edited at 2016-07-13T20:04:08.042Z

Redacted--AC

edited at 2016-07-11T21:59:17.817Z

- Face Analysis Library V1

edited at 2016-07-08T01:02:08.229Z

Redacted--AC

edited at 2016-07-07T22:53:49.998Z

Redacted--AC

Redacted--AC

edited at 2016-07-07T20:26:48.649Z

Redacted--AC

edited at 2016-07-07T16:40:57.427Z

- Used on **Client Only**
- Detect faces snaps or story videos
- Detect facial landmarks (left corner of left eye, right corner of left eye, ..., right corner of mouth, etc), currently detected 68 facial landmarks
- Design:
- Research: [REDACTED]
- QA:
- PM: [REDACTED]

edited at 2016-06-28T04:58:53.359Z

Redacted--AC

edited at 2016-06-27T18:47:02.708Z

- Sheet3 updated

edited at 2016-06-24T00:46:19.840Z

- Snapmoji: Snapmoji Avatar Generation Technology Spec [v1]
- Gallery Search: Gallery Search V4 Spec

- Face Clustering: Face clustering spec

edited at 2016-06-23T22:33:40.384Z

- Design:
- Engineering Research:
- QA:
- PM: V

- Interfaces
- C++ interface
- namespace face {

```
class FaceManager
{
public:

    static FaceManager* get_instance();

    void init();

    static void close();

    void set_facedet_model(const std::string& modelpath);
    void set_landmark_model(const std::string& modelpath);
    void set_feat_model(const std::string& modelpath);
    void set_age_model(const std::string& modelpath);
    void set_expression_model(const std::string& modelpath);
    void set_gender_model(const std::string& modelpath);
    void set_attractive_model(const std::string& modelpath);
    void set_race_model(const std::string& modelpath);
    void set_ghj_model(const std::string& modelpath);

    std::vector<cv::Rect> detect_face(const cv::Mat& inputimg);

    std::vector<cv::Point> detect_landmarks(const cv::Mat& inputimg,
                                           const cv::Rect& facerect);
    cv::Mat frontalize(const cv::Mat& inputimg,
                      const std::vector<cv::Point>& landmarks);

    std::vector<float> predict_attractive(const cv::Mat& inputimg,
                                         const std::vector<cv::Point>& landmarks);

    std::vector<float> predict_race(const cv::Mat& inputimg,
                                   const std::vector<cv::Point>& landmarks,
                                   bool binary=false);

    std::map<std::string, std::vector<float>> >
    predict_ghj(const cv::Mat& inputimg,
               const std::vector<cv::Point>& landmarks);
private:
    FaceManager() {}

    static FaceManager* _instance;
    std::shared_ptr<Feat> _feat_extractor;
    std::shared_ptr<Age> _age_predictor;
    HIGHLY CONFIDENTIAL (COMPETITOR)
```



```
std::shared_ptr<GhjCnn> _ghj_predictor;  
std::shared_ptr<Race> _race_predictor;  
std::shared_ptr<AttractiveCnn> _attractive_predictor;
```

```
}
```

- Snapmoji: Snapmoji Avatar Generation Technology Spec [v1]
- Gallery Search: Gallery Search V4 Spec
- Face Clustering: Face clustering spec

edited at 2016-06-21T22:47:23.897Z

Redacted--AC

- Sheet1_2 updated

edited at 2016-06-17T22:30:31.558Z

- Aligned face
- Face attributes prediction:
- Facial landmark detection
- Face frontalization
- Gender prediction

edited at 2016-06-17T17:27:30.795Z

edited at 2016-06-17T17:16:59.518Z

edited at 2016-06-16T23:31:13.338Z

Redacted--AC

- Feature Overview
- Feature Goals
- Features
- User Stories

Redacted--AC

Use the following template to answer the questions:

- *These questions must be filled out by an **engineer or engineering director two weeks before code freeze** for the target release. All answers should cover any existing and/or planned use of User Data for this feature. Checkout Engineering's User Information Usage Cheatsheet.*

Redacted--AC

- *Once this section is completed and the spec is ready for review....*
- Email [REDACTED] with...

Subject: (Name of this spec) Body: (Target Release and Link to Quip).

- Fill in review request details below:

- Sheet1_2 updated
- Review Results (law / sec)
- *Only **Security and Legal** should fill in this section. If Approved, any future changes will require a new review.*
-
- Sheet3 updated
- Updates
- *All changes to the spec are tracked here. Tag interested parties to notify. Notified should ACK the change in LGTM.*
-
- **NOTE: New content should always be inserted at the top**
- Sheet4 updated

[REDACTED] edited at 2016-06-15T01:01:42.016Z

- Snapmoji: Bitmoji-Snapchat Training Dataset Spec [v1] Snapmoji Avatar Generation Technology Spec [v1]

[REDACTED] edited at 2016-06-14T23:38:17.066Z

- Engineering: [REDACTED]
- QA:
- PM: [REDACTED]
- Snapmoji: Bitmoji-Snapchat Training Dataset Spec [v1] Snapmoji Avatar Generation Technology Spec [v1]

[REDACTED] edited at 2016-06-14T21:48:03.160Z

- Age prediction
- Face feature extraction
- Emotion recognition
- Attractiveness prediction
- Headwear prediction
- (currently only happiness)
- Age: 40
- Race: Asian(prob: 1.0), White(prob:0), Black(prob: 0), Indian(prob:0)
- Attractiveness: (prob: 0.50)
- Emotion Recognition: (Happiness prob: 0.01)
- Headwear: prob: 0.29

[REDACTED] edited at 2016-06-14T16:45:53.413Z

- Face Analysis Lib Library
- Feature Overview
- This spec intends to document the functionalities, usages and its applications .
- Features
- Detect faces snaps or story videos
- Detect facial landmarks (left corner of left eye, right corner of left eye, ..., right corner of mouth, etc), currently detected 68 facial landmarks
- Aligned face
- Face attributes prediction:
- Gender prediction
- Race Prediction
- Age prediction
- Face feature extraction
- Emotion recognition
- Attractiveness prediction
- Headwear prediction
- Owners
- Design:
- Engineering: [REDACTED]
- QA:
- Terms
- Bounding box: The tight region around the face
- Facial landmarks: The most distinctive corners on the face
- prob: Probability
- Functions
- The following image will be used as an example to show the results of different functions.
- Face detection: Detect multiple faces in a snap or frames of videos
- *Image*

- Facial landmark detection: Generate 68 landmarks on the face as in the following example.
- *Image*
-
- Currently, the Face Analysis Library provides the following functions:
- Face alignment
- *Image*
- Attributes Prediction
- Gender: male(prob: 0.94)
- Age: 40
- Race: Asian(prob: 1.0), White(prob:0), Black(prob: 0), Indian(prob:0)
- Attractiveness: (prob: 0.50)
- Emotion Recognition: (Happiness prob: 0.01)
- Headwear: prob: 0.29
- Interfaces
- C++ interface
- namespace face {

```
class FaceManager
{
public:

    static FaceManager* get_instance();

    void init();

    static void close();

    void set_facedet_model(const std::string& modelpath);
    void set_landmark_model(const std::string& modelpath);
    void set_feat_model(const std::string& modelpath);
    void set_age_model(const std::string& modelpath);
    void set_expression_model(const std::string& modelpath);
    void set_gender_model(const std::string& modelpath);
    void set_attractive_model(const std::string& modelpath);
    void set_race_model(const std::string& modelpath);
    void set_ghj_model(const std::string& modelpath);

    std::vector<cv::Rect> detect_face(const cv::Mat& inputimg);

    std::vector<cv::Point> detect_landmarks(const cv::Mat& inputimg,
                                           const cv::Rect& facerect);
    cv::Mat frontalize(const cv::Mat& inputimg,
                      const std::vector<cv::Point>& landmarks);

    std::vector<float> predict_attractive(const cv::Mat& inputimg,
                                         const std::vector<cv::Point>& landmarks);

    std::vector<float> predict_race(const cv::Mat& inputimg,
                                   const std::vector<cv::Point>& landmarks,
                                   bool binary=false);

    std::map<std::string, std::vector<float>> >
    predict_ghj(const cv::Mat& inputimg,
               const std::vector<cv::Point>& landmarks);
private:
```



```

static FaceManager* _instance;
std::shared_ptr<Feat> _feat_extractor;
std::shared_ptr<Age> _age_predictor;
std::shared_ptr<GhjCnn> _ghj_predictor;
std::shared_ptr<Race> _race_predictor;
std::shared_ptr<AttractiveCnn> _attractive_predictor;
};

```

• Privacy Review

Redacted--AC

- Applications
- This section tracks known applications of this library.
- ~~Directory structure~~
- apps/
- frontalize/
- landmark, gender, demo
- core/
- Face detection
- HOG based
- Haar based
- Landmark detection
- Shape regressor
- CNN based
- Frontalization
- 3D based frontalization
- Simple 2D based cropping / rotation
- Gender estimation
- CNN based
- Age estimation
- CNN based
- data/
- Testing data and models
- android/
- Android wrapper
- ios/
- iOS wrapper
- java/
- Java interface
- python/
- Python interface
- test/
- Testing code
-
- Snapmoji: Bitmoji-Snapchat Training Dataset Spec [v1]

edited at 2015-08-28T16:16:47.528Z

- Face Analysis SDK Lib
- apps/
- frontalize/

- landmark, gender, demo
- core/
- Face detection
- HOG based
- Haar based
- Landmark detection
- Shape regressor
- CNN based
- Frontalization
- 3D based frontalization
- Simple 2D based cropping / rotation
- Gender estimation
- CNN based
- Age estimation
- CNN based
- data/
- Testing data and models
- android/
- Android wrapper
- ios/
- iOS wrapper
- java/
- Java interface
- python/
- Python interface
- test/
- Testing code
- apps/

frontalize/

core/

The libface_core library, containing various methods for face detection, landmark detection, frontalization, etc.

data/

The testing data are saved here.

java/

The java programming bindings for libface_core.

python/

The python programming bindings for libface_core.

test/

The testing routines are here.

edited at 2015-08-24T17:43:21.821Z

- Face Analysis SDK
- **Directory structure**

- apps/

frontalize/

core/

The libface_core library, containing various methods for face detection, landmark detection, frontalization, etc.

data/

The testing data are saved here.

java/

The java programming bindings for libface_core.

python/

The python programming bindings for libface_core.

HIGHLY CONFIDENTIAL (COMPETITOR)

SNAP5467024

The testing routines are here.

-

DOCUMENT SLIPSHEET

Bates Number: SNAP5467009-SNAP5467025

BEGATTACH: SNAP5467009

ENDATTACH: SNAP5467029

Custodian: [REDACTED]

Author: [REDACTED]

Filename: FACE ANALYSIS LIBRARY V1.HTML

Create Date:

Last Modified Date:

TNAG BEGBATES:

TNAG ENDBATES:

MAAG BEGBATES:

MAAG ENDBATES:

File Path: /FACE ANALYSIS LIBRARY V1.HTML