Embedded Intelligent Systems Languages and Tools: Dlib

Kevin Hernández Díaz

Dlib-ml: A Machine Learning Toolkit

- General purpose cross-platform open source library
- Written in C++ but with a Python API
- Built to be as portable and easy to use as possible without further installation or configuration or extra packages needed
  - Works on OS X, MS Windows, Linux, Solaris, the BSDs, and HP-UX

- King, D. E. (2009). Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, *10*, 1755-1758.

HALMSTAD UNIVERSITY

Library content categories

- Really general purpose:
  - Algorithms (maths and other that does not fit other categories)
  - API Wrappers
  - Bayesian Nets
  - Compression
  - Containers
  - Graph Tools
  - **Image Processing**
  - Linear algebra
  - **Machine Learning**
  - Metaprogramming
  - Miscellaneous
  - Networking
  - Optimization
  - Parsing

HALMSTAD UNIVERSITY

Image Processing Tools

- Basic Pixel functions and structures
  - assign_pixel, bgr_pixel, rgb_pixel
- Image I/O
  - Load_image, load_png, save_bmp…
- Object Detection
  - Find_candidate_object_locations, remove_unobtainable_rectangles, traditional object detector based on hog and pyramidsLinear algebra

HALMSTAD UNIVERSITY

Image Processing Tools

- Feature Extraction
  - HOG, lbp, histograms, find keypoints, SURF, shape_predictor…
- Edges and Thresholds
  - Find_bright_lines, find_dark_lines, sobel, hough_transform…
- Morphology
  - Close, open, dilation, erosion, union, find_line_endpoints, label_connected_blobs…

HALMSTAD UNIVERSITY

Image Processing Tools

- Filtering
  - Gaussian_blur, max_filter, sum_filter, apply custom filters…
- Scaling and Rotating
  - Flip images, rotate, resize, create image pyramids, random jittering
- Visualization
  - Draw lines, rectangles, circles, keypoints, heatmaps, HOG features… On images
- Miscellaneous
  - Wrapper for OpenCV images, histogram equalization, image padding…

HALMSTAD UNIVERSITY

Machine Learning Tools

- Binary and Multiclass classification
  - Mostly SVMs, but also tools like one_vs_all_trainer, one_vs_one_trainer
- Regression
  - RF, MLP, SVM...
- Clustering
  - Chinese_whispers, kkmeans, nearest_center...
- Unsupervised and Semi-Supervised, and Reinforcement Learning

HALMSTAD UNIVERSITY

Machine Learning Tools

- Other tools
  - Feature selection, validation, data IO...

- Deep Learning
  - Basic tools that allows to create, load, train NNs, and to create predictions with them
  - Each network is form by three components:
    - Input_layer, Computational layers, and a loss layer.

HALMSTAD UNIVERSITY

Conclusions

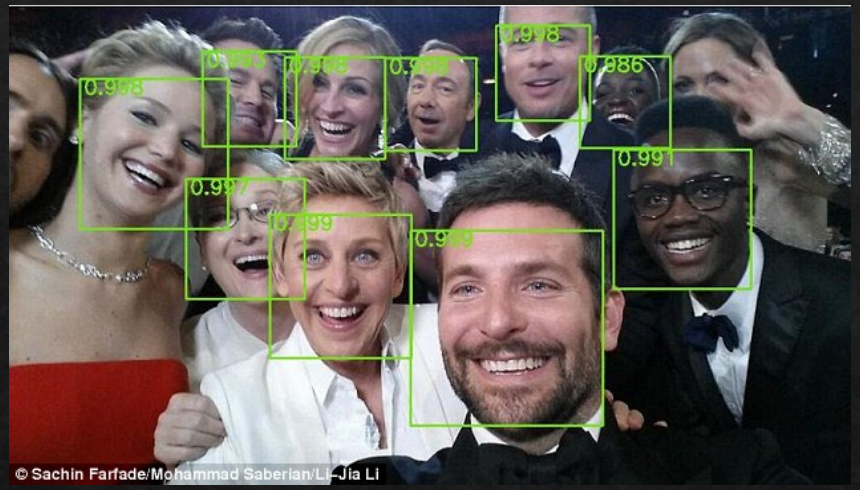
- Pros:
  - Self-contained. Capable of coding whole apps with just one light library without extra packages.
  - Many different functionalities
  - Compartamentalzied and transparent with the methods
- Cons:
  - «Jack of all trades, master of none.» Decent, but probably not the best in any form
  - Bloated library. Many similar functions with slightly different behaviors

Lab case: Face Biometrics

- Face detection
- Landmark detection
- Face alignment
- Face data augmentation
- Face recognition
- Applications (drowsiness detection)

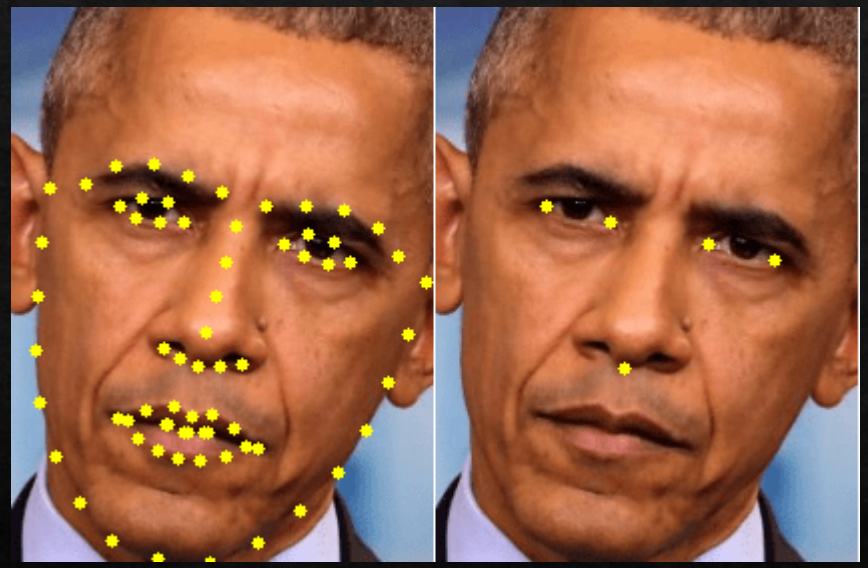
HALMSTAD UNIVERSITY

Face Detection



- Is there a face in the image? How many, and where?
- Dlib provides 2 different face detectors
  - One is based on traditional object detector method (HOG+classifier over pyramid image)
    - **Detector = dlib.get_frontal_face_detector()**
  - The other is based on CNNs
    - Detector = dlib.cnn_face_detection_model_v1("model_file.dat")
- Dets = Detector(img,1)

Landmark Detection



- Extract the landmark points of the face in the image
- Dlib provides 2 different face landmark detectors/predictor
  - One is just to extract 5 points. Both eye corners and nose center
    - **sp = dlib.shape_predictor(shape_predictor_5_face_landmarks.dat)**
  - The other is 68 points across the face
    - **sp = dlib.shape_predictor(shape_predictor_68_face_landmarks.dat)**
- Landmarks = sp(img, detected_face)

Face alignment

- Align faces to improve recognition
  - Detect faces
  - Extract landmarks
  - align
- You can implement your own alignment method
- Dlib provides some functions for this
  - Aligned = dlib.get_face_chips(img, landmarks_per_face, other parameters)

Data augmentation

- Dlib provides out-of-the-box functions to perform data augmentation. Doing random rotation, scaling, translation and flips.

- Aug_data = dlib.jitter_image(image, num_jitters)

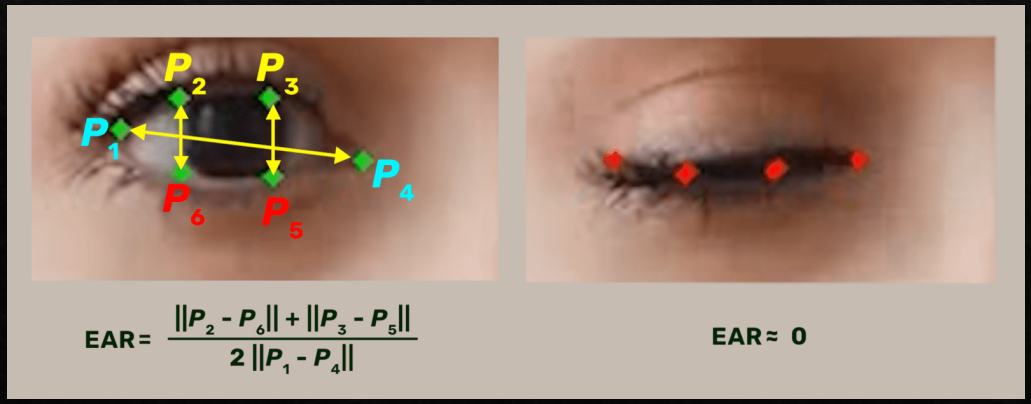
Face recognition

Obama

- Recognition process
  - Detect faces
  - Extract landmarks
  - Align
  - Perform face recognition
- You can implement your own recognition method with dlib functions:
  - Feature extraction methods: LBP, HOG, SURF...
  - Classification methods: SVMs, NN...
- Dlib provides a especific face recognition CNN model based on resnet
  - Facerec = dlib.face_recognition_model_v1(model.dat)

Application drowsiness detection

- A simple approach to detect drowsiness while driving is monitoring eye closure.
- We saw/will see how to extract landmark points that include the eye shape
- There is a metric called Eye Aspect Ratio (EAR) as shown below
- Monitoring how many frames in a video the eye is closed or mostly closed can help detect drowsiness



Thanks!
Questions?