Through the eyes of RoboFace

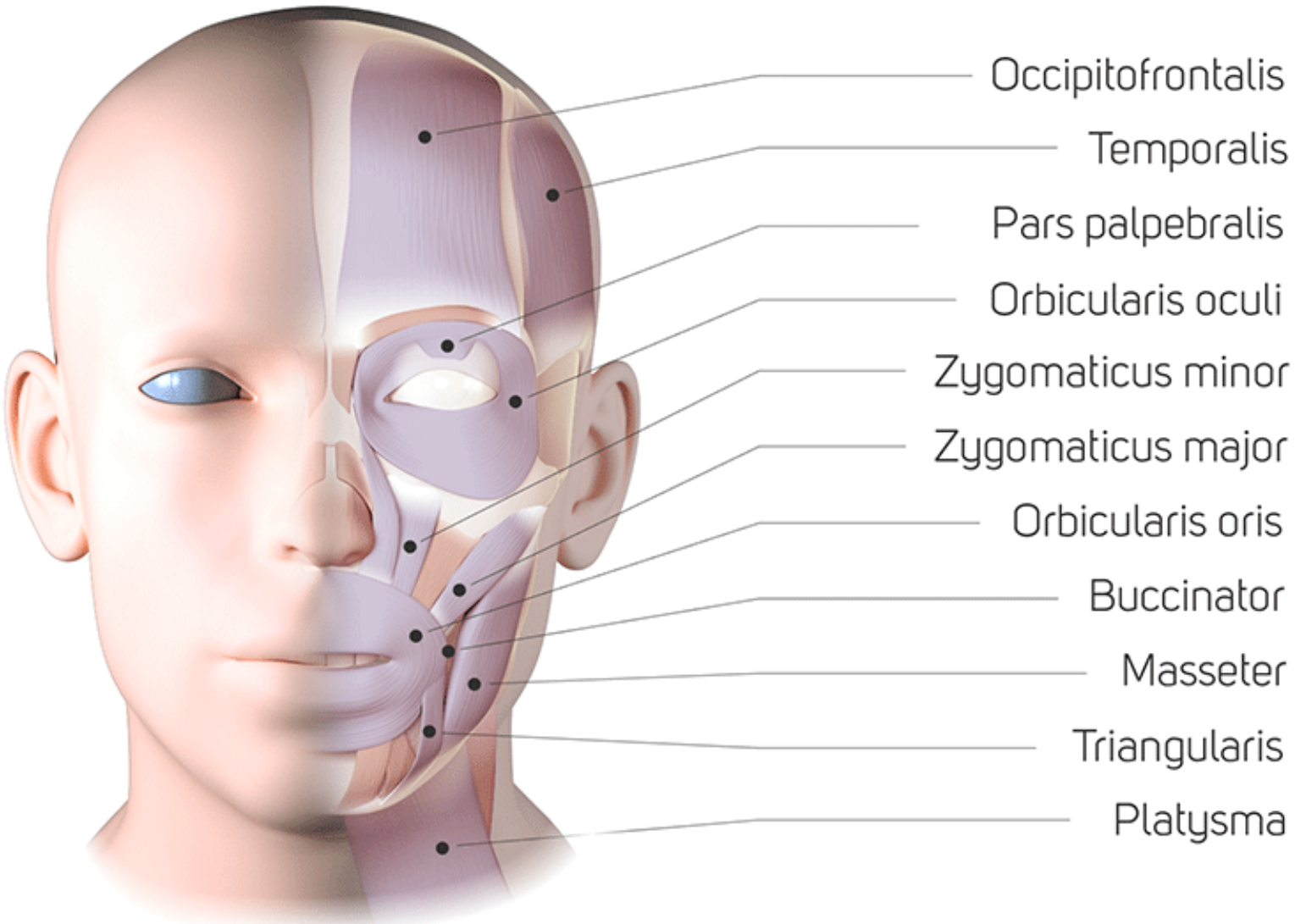
Robotics practical – final project

Kevin Kiefer, Letiția Elena Pârcălăbescu

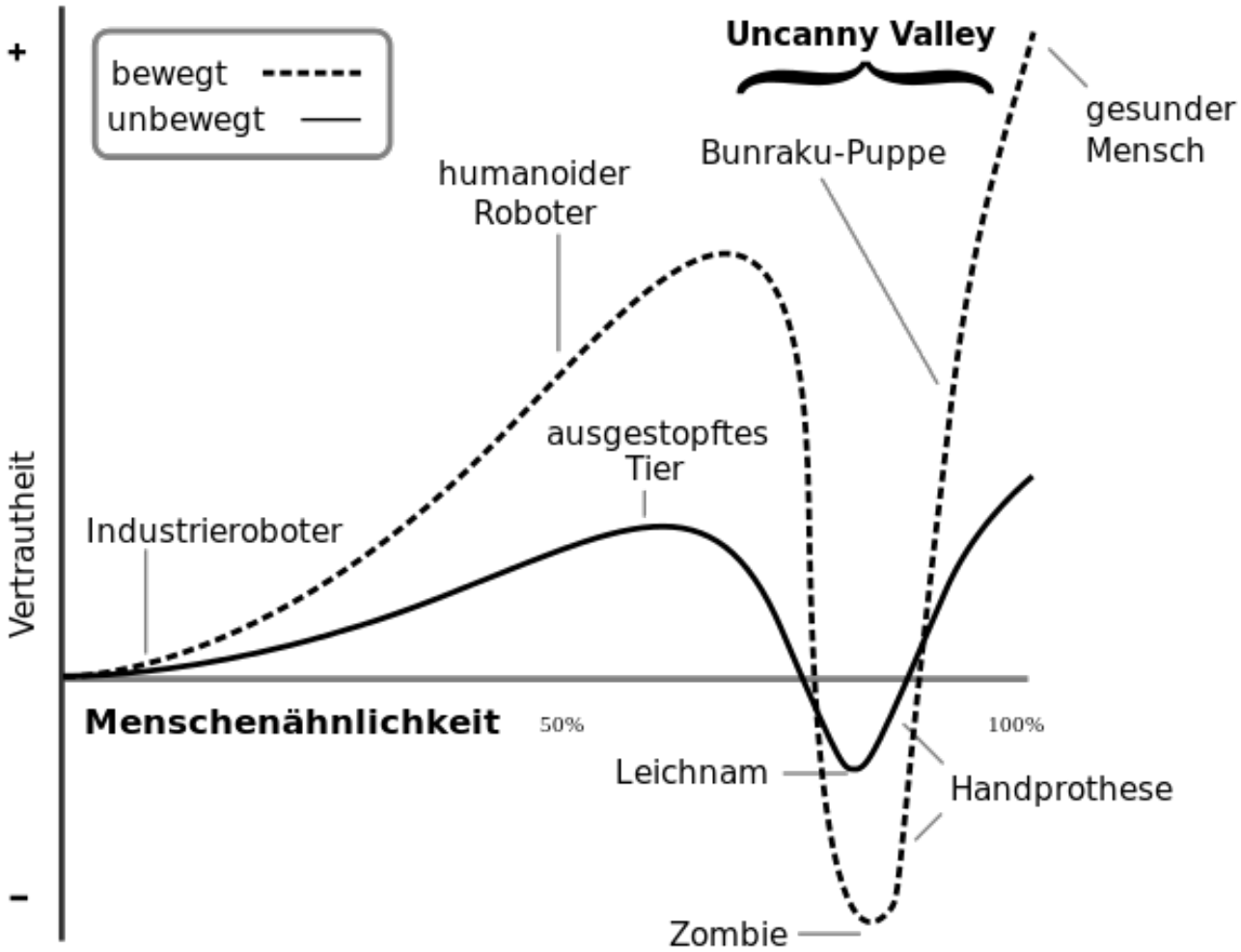
the 28th of March 2017

Structure

HumanFace


Occipitofrontalis
Temporalis
Pars palpebralis
Orbicularis oculi
Zygomaticus minor
Zygomaticus major
Orbicularis oris
Buccinator
Masseter
Triangularis
Platysma

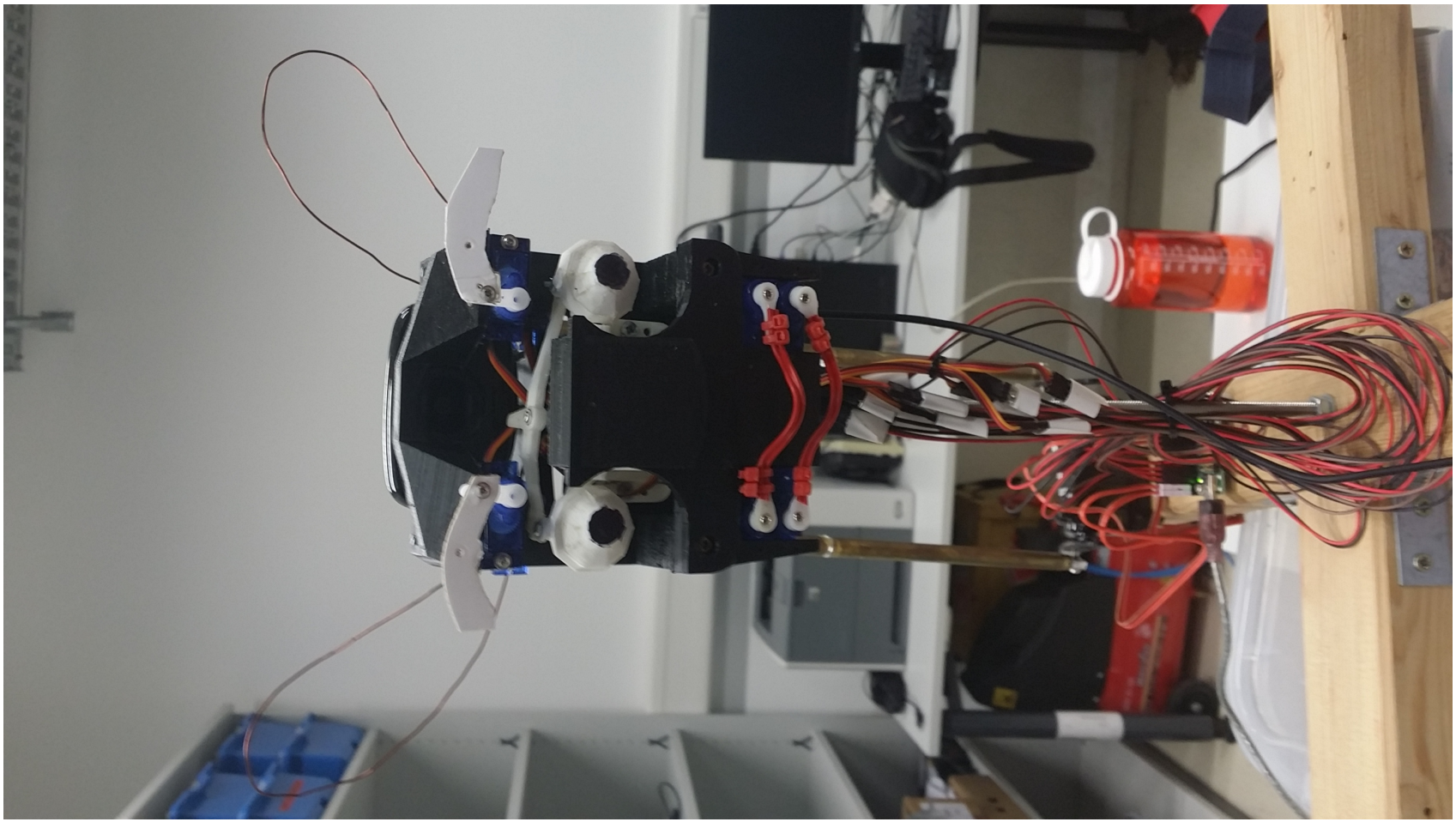
Uncanny Valley



Uncanny Valley

RoboFace

Hardware



RoboFace

Hardware

RoboFace
Software

1. dependecies: pololu-usb-sdk, RapaPololuMaestro, boost(python), python(2 or 3)
2. low-lvl code implemented in c++
3. python api build on top
4. main workflow implemented in python

RoboFace
Software

```
template<size_t N>
class ServoConfig
{
public:
    ServoConfig(
        const std::array<int,N> & servoChannel,
        const std::array<int,N> & servoPos
    );
    // getter & setter
private:
    std::array<int, N> servoChannel_;
    std::array<int, N> servoPos_;
};
```

RoboFace
Software

```
template<size_t NUM_SERVOS>
struct ServoConstraints
{
public:
    ServoConstraints(
        int min, int max,
        const std::array<int, NUM_SERVOS> & channels
    );

    bool isValidChannel(int channel) const;
    bool isValidPosition(int pos) const;

    // isValidChannelArray & isValidPositionArray

    template<size_t N>
    typename std::enable_if<(N <= NUM_SERVOS), bool>::type
    isValidConfig(const ServoConfig<N> & config) const;

private:
    std::array<int, NUM_SERVOS> channels_;
    int minPos_;
    int maxPos_;
};
```

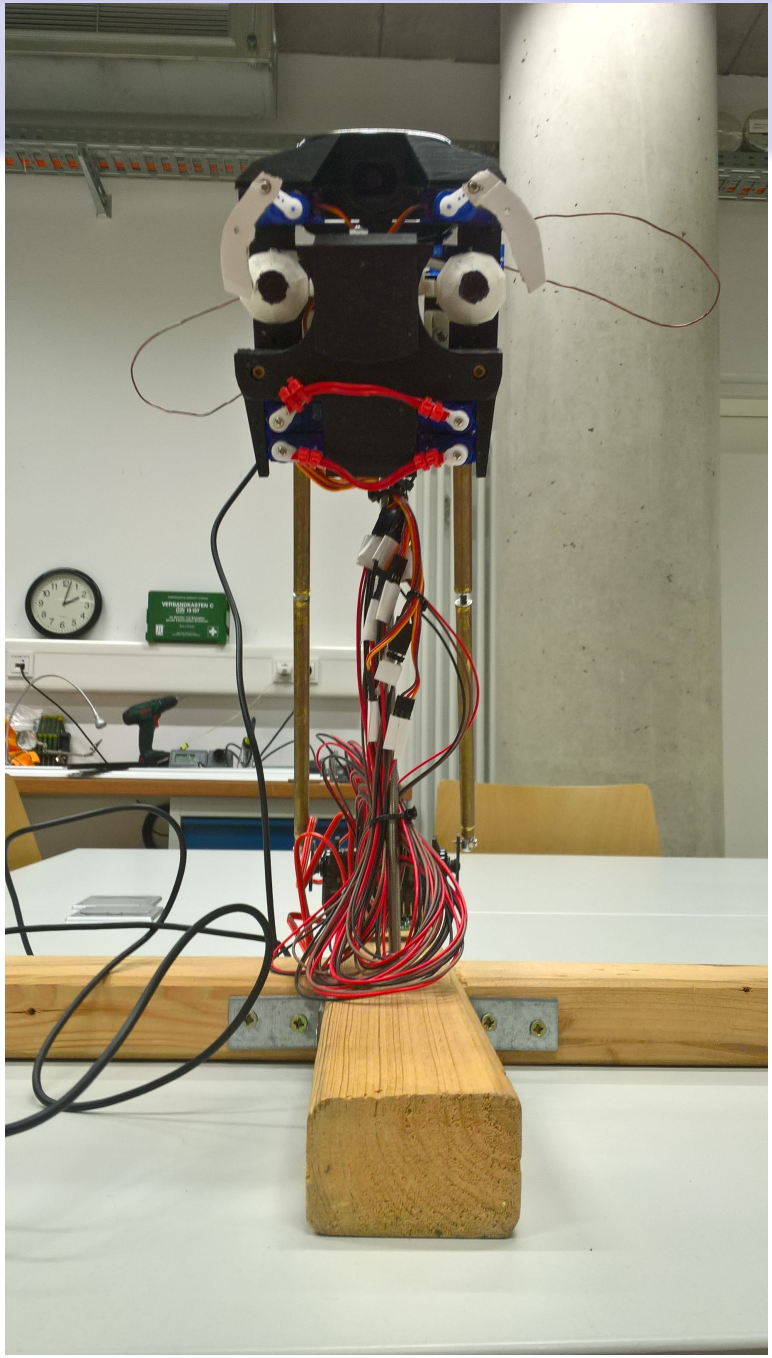
RoboFace

Software

```
class Face
{
public:
    Face(
        int x_len = 640 , int y_len = 480,
        float x_weight = 0.5f, float y_weight = 0.5f,
        const std::string & dev = "/dev/ttyACM0"
    );

    template<size_t N>
    void applyConfig(const ServoConfig<N> & config)

    void happy(bool moveHead = true)
    // more emotions

    void moveHeadX(int x)
    void moveHeadY(int y)
    void moveHead(int x, int y)
private:
    // ...
};
```

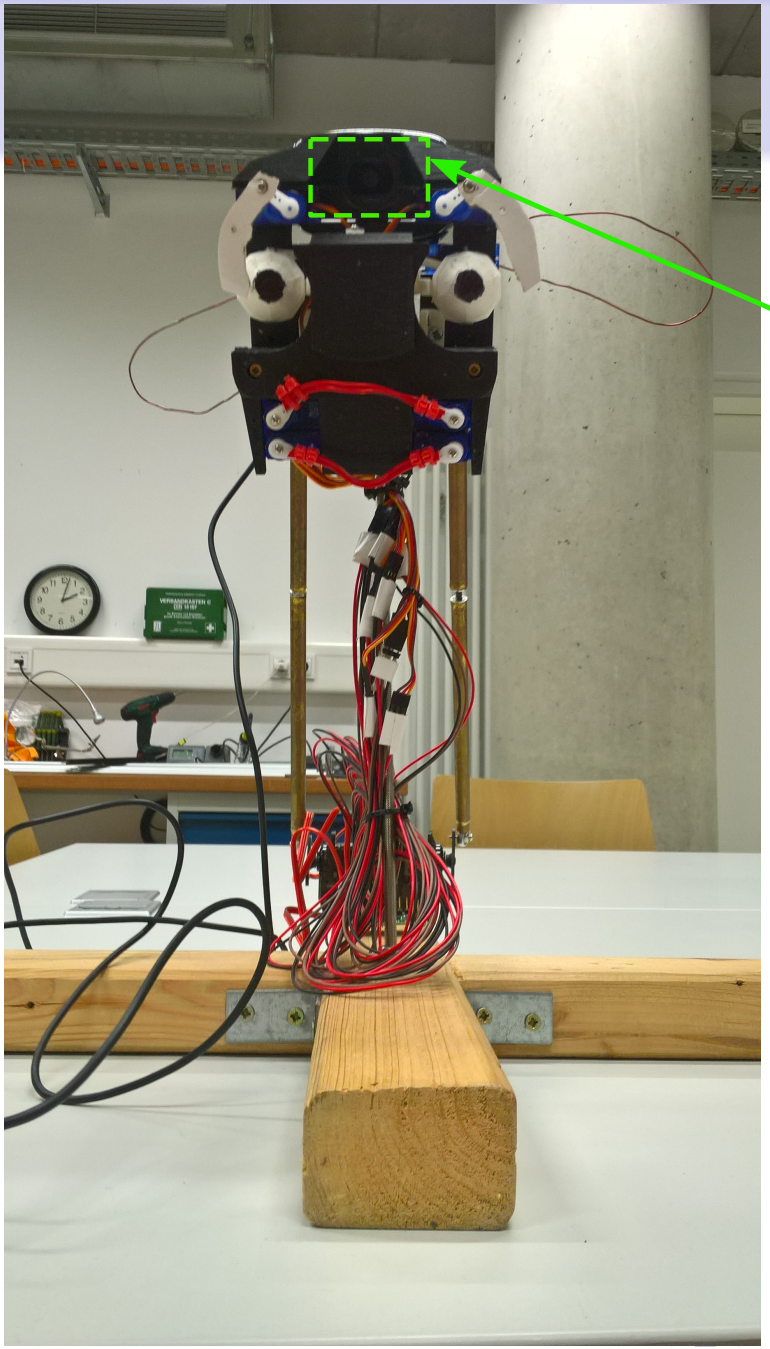
RoboFace

Deep Vision Problem



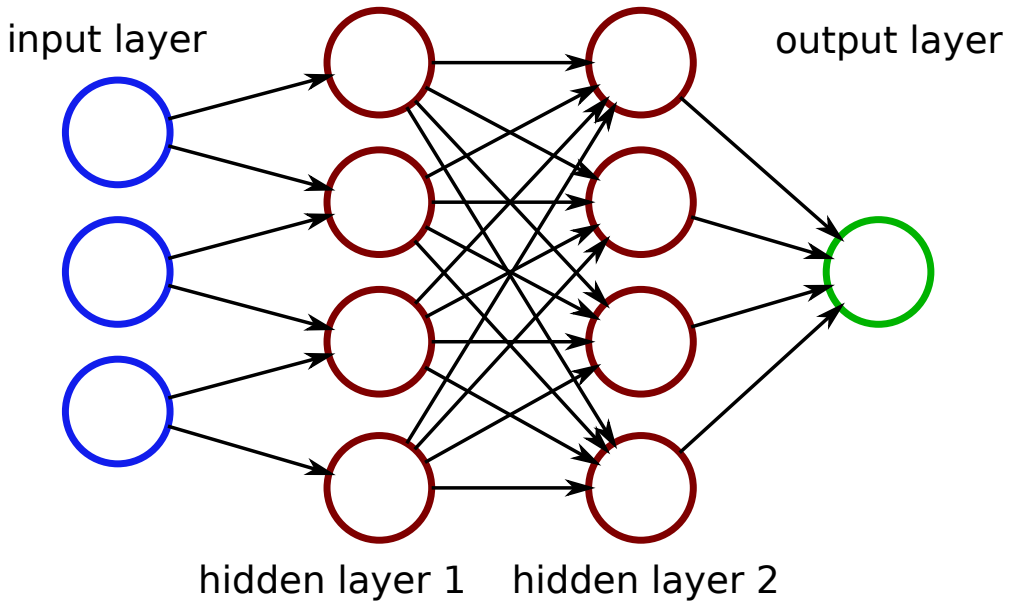
RoboFace

Deep Vision Problem



Deep Vision

Short Intro into Neural Networks



CelebA Dataset web (2016)

202.599 images, 10.177 identities, 40 attributes per image

Large-scale CelebFaces Attributes (CelebA) Dataset

Ziwei Liu Ping Luo Xiaogang Wang Xiaoou Tang
Multimedia Laboratory, The Chinese University of Hong Kong

CelebA Dataset

Sample Images

CelebA Dataset

Attribute Selection

chose 13 labels out of 40 available

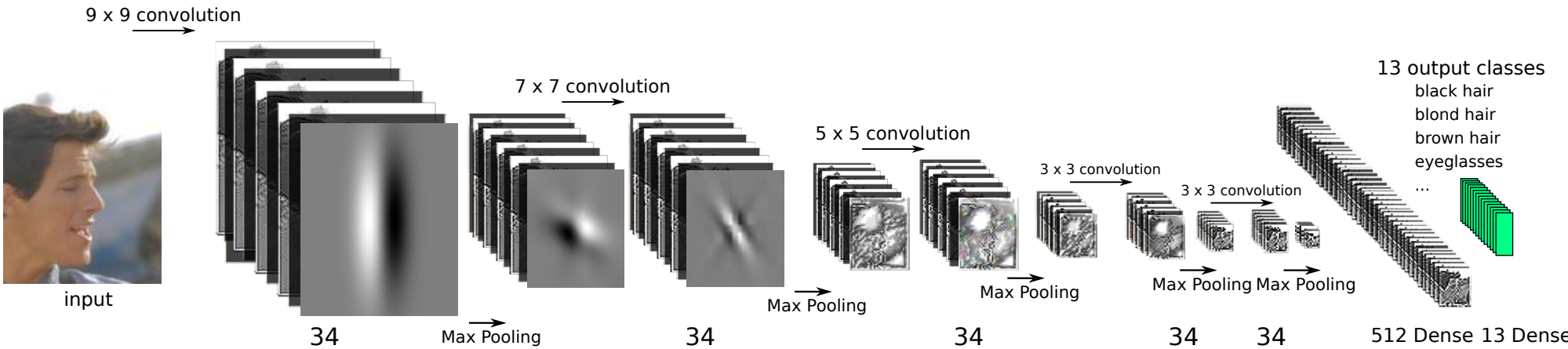
1. Black Hair
2. Blond Hair
3. Brown Hair
4. Eyeglasses
5. Gray Hair
6. Male
7. Mouth Slightly Open
8. No Beard
9. Smiling
10. Straight Hair
11. Wavy Hair
12. Wearing Earrings
13. Wearing Lipstick

Own Approach

Architecture

1. 32 of neurons in each convolutional layer
2. Activations: Relu
3. 9 x 9 convolution $\rightarrow$ Max Pooling
4. 7 x 7 convolution $\rightarrow$ Max Pooling
5. 5 x 5 convolution $\rightarrow$ Max Pooling
6. 3 x 3 convolution $\rightarrow$ Max Pooling
7. 3 x 3 convolution $\rightarrow$ Max Pooling
8. Dropout(0.25)
9. 512 Dense
10. Dropout(0.5)
11. 13 Dense
12. Sigmoid
13. Binary Crossentropy
14. Adadelta optimiser
15. Overall: 125,709 parameters

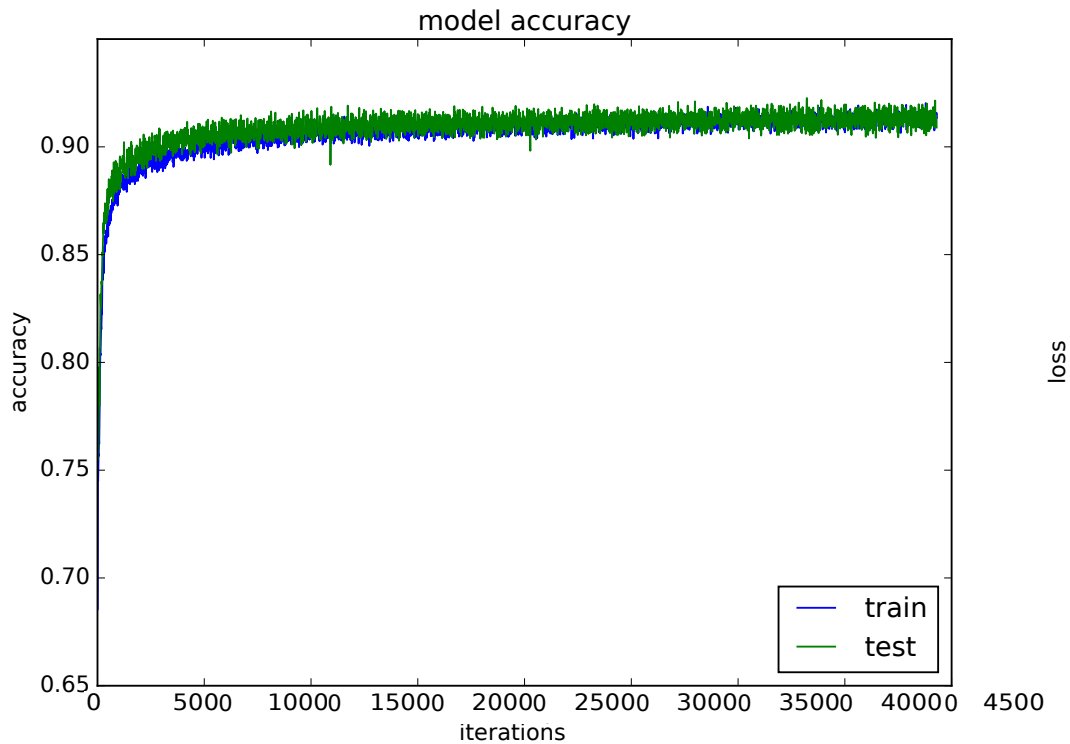
Own Approach
Architecture

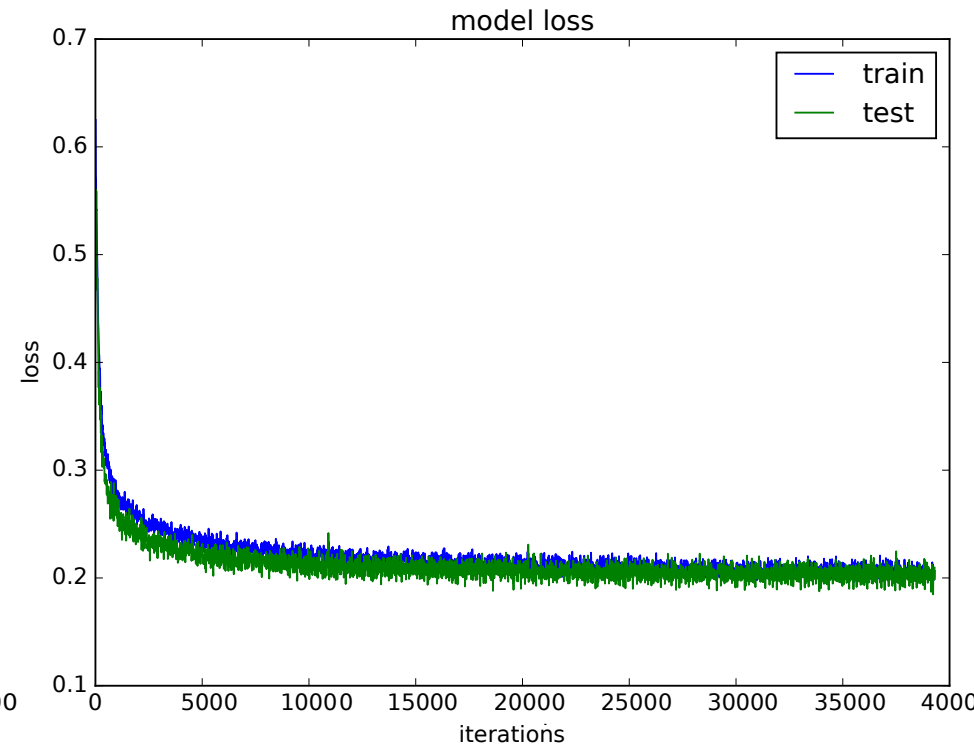


Own Approach

How to not do it!

Without data normalisation: beautiful training curve, BUT...





Own Approach

How to not do it!

... BUT no generalisation capability at all!

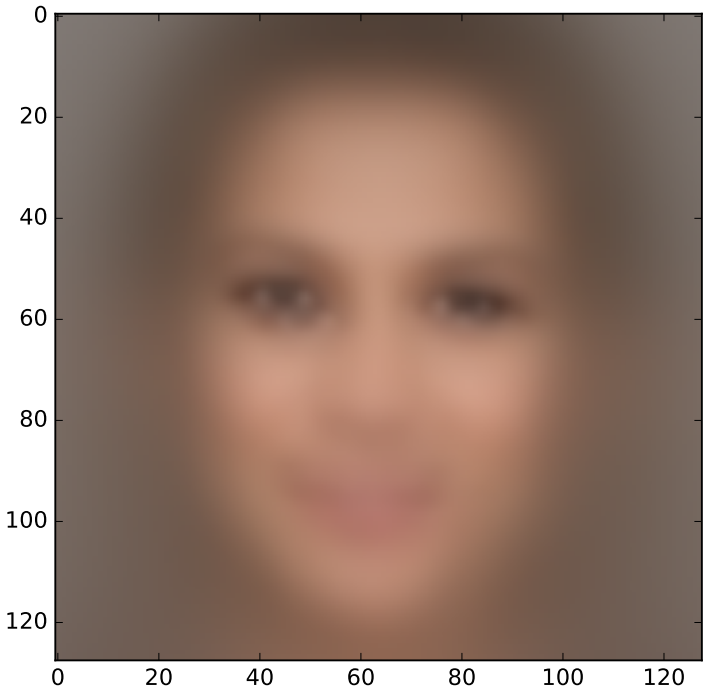
The predicted classes on the real world images from the robot are:

Male (0.89), No beard (0.76)

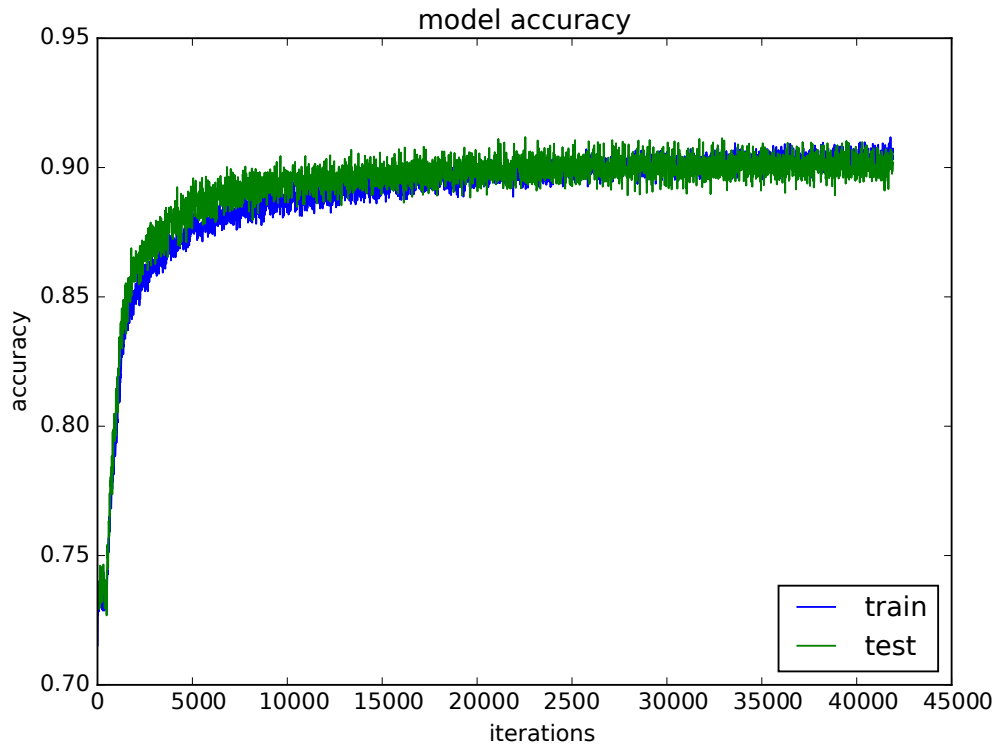
Data normalisation

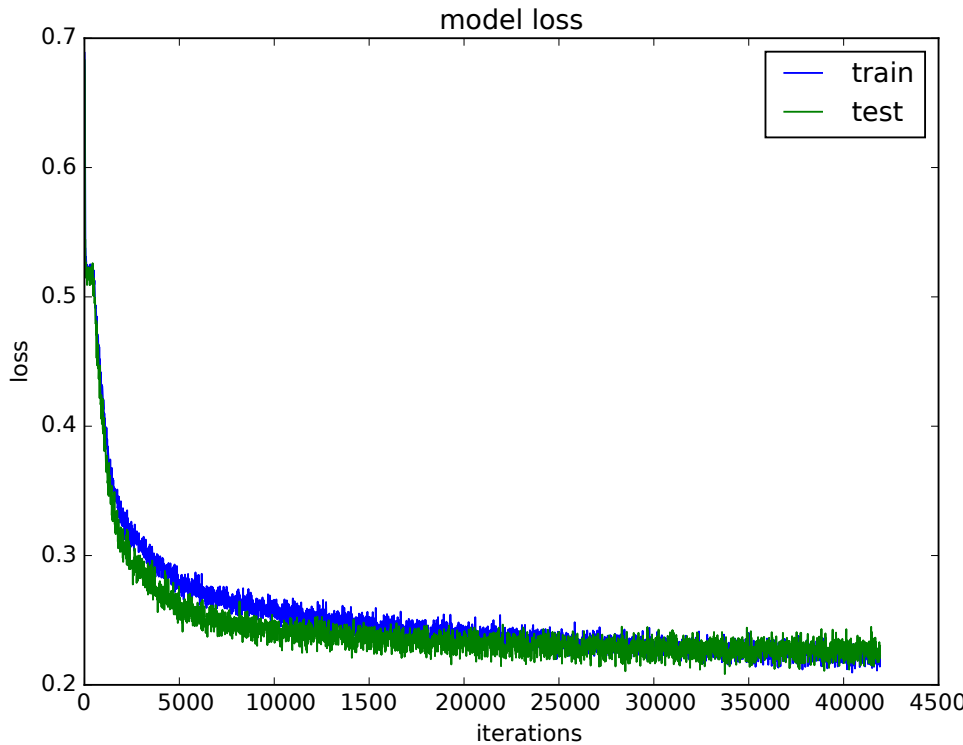
How to really do it!

1. center the eyes
2. resize images to have the same inter ocular distance
3. rotate image to make the inter ocular line look horizontal
4. resize image to 128 x 128 pixels
5. subtract the mean face



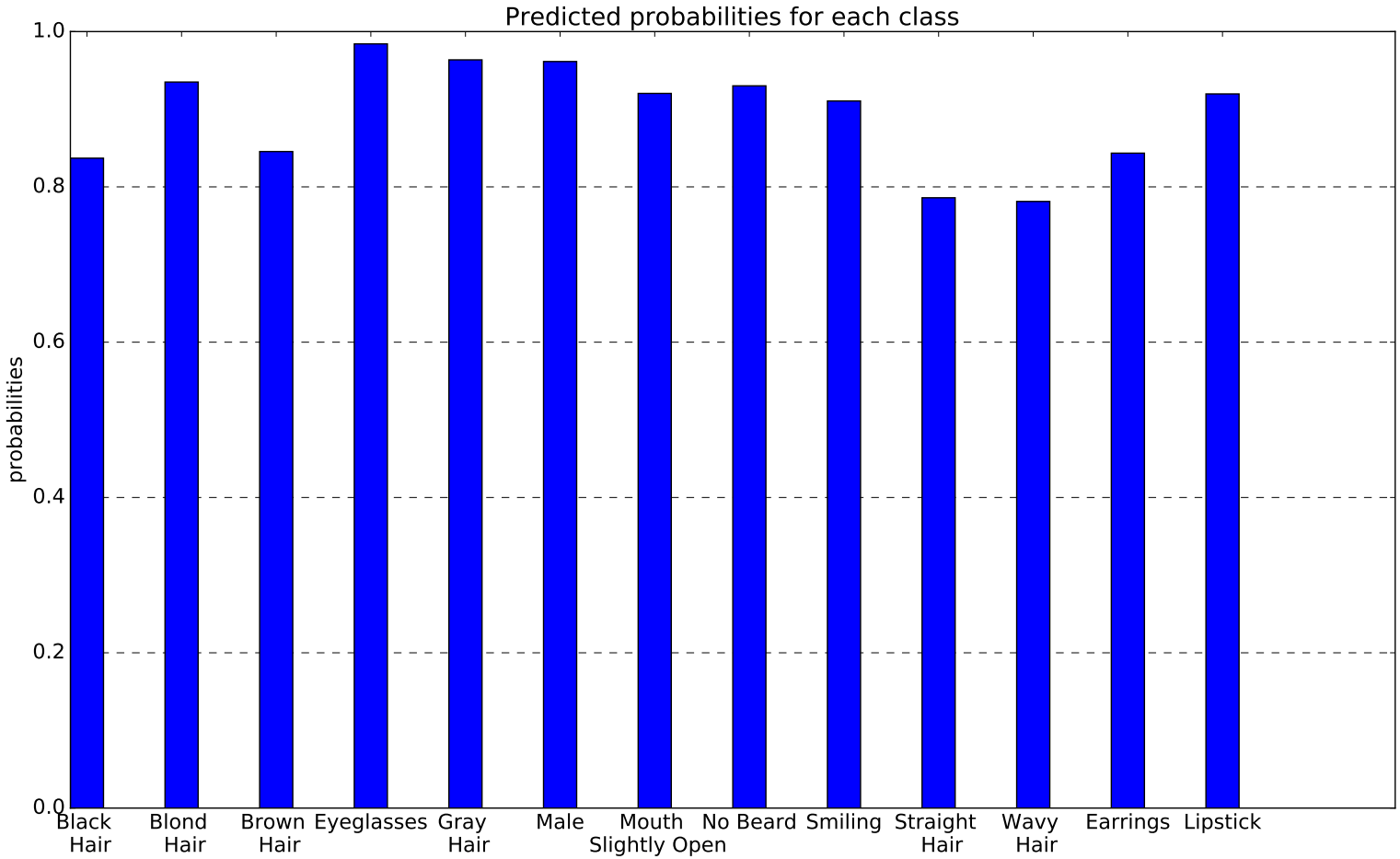
Own Approach
Training Curves





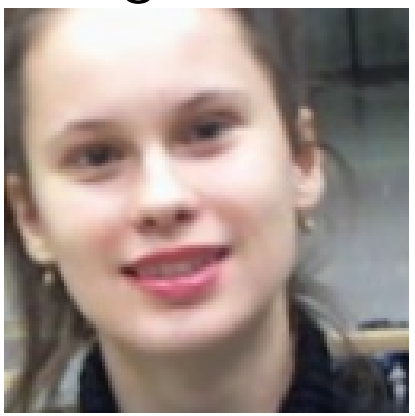
Own Approach
Final Accuracy per Class

overall accuracy of 90%

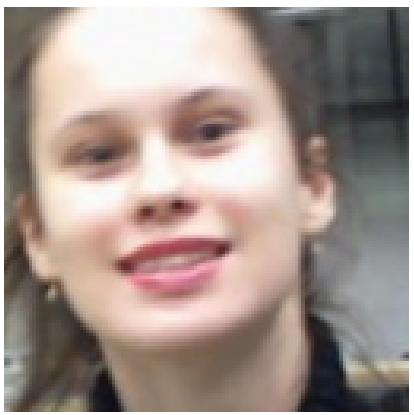


Exemplary Results

'Black Hair', 'Mouth Slightly Open', 'No Beard', 'Smiling', 'Wearing Earrings', 'Wearing Lipstick'



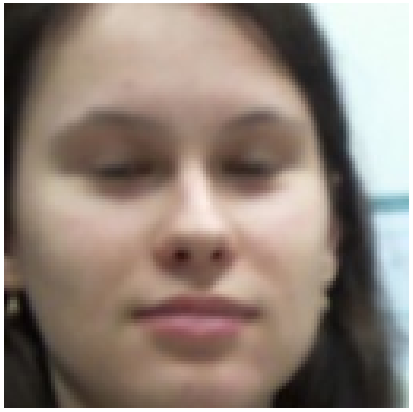
'Black Hair', 'Mouth Slightly Open', 'No Beard', 'Straight Hair', 'Wearing Lipstick'



Exemplary Results

'Black Hair', 'Mouth Slightly Open', 'No Beard', 'Smiling', 'Straight Hair', 'Wearing Earrings', 'Wearing Lipstick

'Black Hair', 'Male', 'Straight Hair' – only 2 out of 45 examples.



Future Potential

- hardware
- speech recognition
- data acquisition

References

Celeba website, 2016. URL
http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.