Let’s talk about Artificial Intelligence and Robotics!

Silvia Santano and Pepper

München, 11.04.2018
Silvia Santano

› Robotic Applications developer
› Programming robots since I was 12
› At inovex since June 2016
Agenda

› Computer Vision
› Image Recognition
› Pepper
   › Softbank Robotics
   › Characteristics
   › Usecases
› Computer Vision
› External services
› On-device CV with CNNs
Agenda

› Computer Vision
  › Image Recognition
  › Pepper
    › Softbank Robotics
    › Characteristics
    › Usecases
  › Computer Vision
  › External services
  › On-device CV with CNNs
Automatic extraction, analysis and understanding of information from images
Humans can recognize objects in images with little effort despite of huge variations
For computers, this is still a challenge...
Agenda

› Computer Vision
› Image Recognition
› Pepper
   › Softbank Robotics
   › Characteristics
   › Usecases
› Computer Vision
› External services
› On-device CV with CNNs
Image Recognition

Main subfields:

› Classification
  ›› Object detection
  ››› Semantic segmentation
Image Recognition

Main subfields:

› Classification
›› Object detection
››› Semantic segmentation
Image Recognition

Main subfields:

› Classification
›› Object detection
››› Semantic segmentation
Image Recognition

Main subfields:

› Classification
  ›› Object detection
  ››› Semantic segmentation

› Identification
Agenda

› Computer Vision
› Image Recognition
› Pepper
   › Softbank Robotics
   › Characteristics
   › Usecases
› Computer Vision
› External services
› On-device CV with CNNs
Softbank Robotics

› Leader in Humanoid Robotics
› Previously: Aldebaran Robotics
› Headquartered in Paris
› Creator of NAO, Pepper and Romeo
› The robots are used in > 70 countries
› Fields: research, education, retail, healthcare, tourism, hospitality or entertainment, SoftBank

› Ecosystem of Certified Partners
Softbank Robotics
Agenda

› Computer Vision
› Image Recognition
› Pepper
   › Softbank Robotics
   › Characteristics
   › Usecases
   › Computer Vision
   › External services
   › On-device CV with CNNs
Pepper: Technical Characteristics

- 120 cm
- 28 kg
- 20° of freedom
- 360° movement
- Wi-Fi connection
- 12h of autonomy

- Tactile sensors (Head, Hands and Bumpers)
- 4 directional microphones
- Cameras 3D and HD
- Touch screen
## Pepper: Technical Characteristics

### Processor
- **Atom E3845**

### CPU
- **Quad core**

### Clock speed
- **1.91 GHz**

### RAM
- **4 GB DDR3**

### OS
- **Nao QI OS**

### Sensors
- 2 HD Cameras (OV5640)
- 1 3D Sensor (ASUS XTION)
- 4 Microphones
- 3-axis Gyrometer + 3-axis Accelerometer
- 6 laser line generators
- 2 Infra-Red sensors
- 2 ultrasonic sensors
- 3 tactile sensors
- 3 bumpers
- 20 Motors and actuators

### Tablet

<table>
<thead>
<tr>
<th>Processor</th>
<th>ARM Cortex-A7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Quad core</td>
</tr>
<tr>
<td>Clock speed</td>
<td>1.3 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>DDR3 SDRAM 1GB (512MB * 2)</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>32GB (eMMC)</td>
</tr>
<tr>
<td>OS</td>
<td>Android 6.0</td>
</tr>
<tr>
<td>Dimensions</td>
<td>246 x 175 x 14.5 mm</td>
</tr>
<tr>
<td>Display</td>
<td>Type: IPS, Resolution: 1280*800 Color: 24bit true color</td>
</tr>
<tr>
<td>Touch Panel</td>
<td>Capacitive Multi-Touch (5 point)</td>
</tr>
<tr>
<td>Sensors</td>
<td>Light illumination, Acceleration Gyro, Geomagnetic</td>
</tr>
</tbody>
</table>
Agenda

› Computer Vision
› Image Recognition
› Pepper
  › Softbank Robotics
  › Characteristics
› Usecases
› Computer Vision
› External services
› On-device CV with CNNs
So, what can this ‘Pepper’ actually do?

Pepper is:

› Proactive
› Attractive
› Interactive
› Emotional and empathetic
› Connected
› Customisable
So, what can this ‘Pepper’ actually do?

Usecases:

› Welcoming
› Informing and recommending products
› Guiding
› Attracting
› Improving customer knowledge
› Entertaining
› ...

Images: Softbank Robotics
Pepper is still a work in progress...
How to program Pepper

› Choregraphe & Python
› Python SDK
› C++ SDK
› Javascript (Tablet)
› Soon: Android (still reduced function set)
› QiChat (Dialogs)
How to program Pepper
Choregraphe
How to program Pepper
QiChat

```javascript
// Volume
## down
u:[
    "{-can_you} {"ein bisschen" etwas} leiser [sprechen reden]"
    "sprich {"ein bisschen" etwas} leiser"
    "Dreh die Lautstärke runter"
    "sprich nicht so laut"
    "du sprichst zu laut"
]
^gotoReactivate(decrease_volume)
u:($empty) %decrease_volume
^call(ALVolumeSlider.decreaseVolume()) $Demo/back=1
  c1:(false) es tut mir leid, das ist das Minimum
  c1:(true) okay ich spreche jetzt leiser

  u2:[
      nochmal
      mehr
      "noch {"ein bisschen" etwas} mehr"
      "immer noch zu laut"
  ]
  ^gotoReactivate(decrease_volume)
```
Agenda

› Computer Vision
› Image Recognition
› Pepper
   › Softbank Robotics
   › Characteristics
   › Usecases
› Computer Vision
› External services
› On-device CV with CNNs
Computer Vision with Pepper

› Face Detection and People Tracking
› Face Learning and Recognition
› People Characteristics Perception
Computer Vision with Pepper
People Characteristics Perception

PeoplePerception/Person/<ID>/AgeProperties
PeoplePerception/Person/<ID>/ExpressionProperties
PeoplePerception/Person/<ID>/GenderProperties
PeoplePerception/Person/<ID>/SmileProperties
PeoplePerception/Person/<ID>/FacialPartsProperties
PeoplePerception/Person/<ID>/Distance
PeoplePerception/Person/<ID>/IsFaceDetected
PeoplePerception/Person/<ID>/IsVisible
PeoplePerception/Person/<ID>/NotSeenSince
PeoplePerception/Person/<ID>/PresentSince
PeoplePerception/Person/<ID>/RealHeight
PeoplePerception/Person/<ID>/ShirtColor
Computer Vision with Pepper

› Face Detection and People Tracking
› Face Learning and Recognition
› People Characteristics Perception
› Emotion Recognition
Computer Vision with Pepper

Emotion Recognition Module

› Data sources:
  › Expression and smile
  › Acoustic voice emotion analysis
  › Head angles
  › Touch sensors
  › Semantic analysis from speech
  › Sound level and energy level of noise
  › Movement detection

Valence
Attention Level
Smile
Expression
{
  “calm"
  “anger"
  “joy"
  "sorrow"
  "laughter"
  "excitement"
  “surprise"
}
(Real values normalized)
Computer Vision with Pepper

› People Tracking
› Face Detection, Learning and Recognition
› People Perception
› Emotion Recognition
› Vision Recognition
› Barcode Reader
Computer Vision with Pepper

People Perception
Emotion Recognition
Face Detection and Recognition
Agenda

› Computer Vision
› Image Recognition
› Pepper
   › Softbank Robotics
   › Characteristics
   › Usecases
› Computer Vision
› External services
› On-device CV with CNNs
External services integration

Google’s Machine Learning Cloud Vision API

› Machine learning service with pre-trained models
› JSON REST API / client libraries (C#, GO, Java, Node.js, PHP, Python, Ruby)

Explicit Content Detection
Logo Detection
Label Detection
Landmark Detection
Optical Character Recognition
Face Detection
Image Attributes
Web Detection
External services integration

Google’s Machine Learning Cloud Vision API: LABELS

<table>
<thead>
<tr>
<th>Faces</th>
<th>Labels</th>
<th>Web</th>
<th>Properties</th>
<th>Safe Search</th>
<th>JSON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Image](DSC_2074.JPG)

- Technology: 93%
- Room: 88%
- Shoulder: 82%
- Arm: 73%
- Robot: 68%
- Machine: 62%
- Product: 62%
- Electronic Device: 59%
- Girl: 56%

https://cloud.google.com/vision/
External services integration

Google’s Machine Learning Cloud Vision API: LOGO DETECTION
Good, but show me the code...
External services integration

Google’s Machine Learning Cloud Vision API: LABELS

def detect_labels(path):
    """Detects labels in the file."""
    client = vision.ImageAnnotatorClient()

    with io.open(path, 'rb') as image_file:
        content = image_file.read()

    image = types.Image(content=content)

    response = client.label_detection(image=image)
    labels = response.label_annotations
    print('Labels:')

    for label in labels:
        print(label.description)

client libraries (C#, GO, Java, Node.js, PHP, Python, Ruby)
External services integration
Google’s Machine Learning Cloud Vision API: LABELS

POST https://vision.googleapis.com/v1/images:annotate?key=YOUR_API_KEY

```
{
  "requests": [
    {
      "image": {
        "content": "/9j/7QBEUGhvdG9zaG9w...base64-encoded-image-content...fXNWzvDE",
      },
      "features": [
        {
          "type": "LABEL_DETECTION"
        }
      ]
    }
  ]
}
```

JSON REST API
External services integration

Google’s Machine Learning Cloud Vision API

Logo Detection
Label Detection
Optical Character Recognition
Web Detection
Emotion Detection
External services integration
Microsoft Cognitive Services

› Machine learning service with pre-trained models
› JSON REST APIs / client libraries (C#, Android, Swift)

Computer Vision API:
  • Analyze Image
  • Optical Character Recognition
  • Handwritten Text Detection
Face API
Emotion API
External services integration
Microsoft Cognitive Services: COMPUTER VISION API

<table>
<thead>
<tr>
<th>FEATURE NAME:</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>{ &quot;tags&quot;: [ &quot;person&quot;, &quot;indoor&quot;, &quot;woman&quot;, &quot;holding&quot;, &quot;man&quot;, &quot;front&quot;, &quot;table&quot;, &quot;white&quot;, &quot;black&quot;, &quot;standing&quot;, &quot;young&quot;, &quot;cake&quot;, &quot;playing&quot;, &quot;dog&quot;, &quot;room&quot;, &quot;plate&quot;, &quot;remote&quot;, &quot;kitchen&quot; ], &quot;captions&quot;: [ { &quot;text&quot;: &quot;a woman standing in front of a cake&quot;, &quot;confidence&quot;: 0.75387466 } ] }</td>
</tr>
<tr>
<td>Tags</td>
<td>[ { &quot;name&quot;: &quot;wall&quot;, &quot;confidence&quot;: 0.996851742 }, { &quot;name&quot;: &quot;person&quot;, &quot;confidence&quot;: 0.996797 }, { &quot;name&quot;: &quot;indoor&quot;, &quot;confidence&quot;: 0.973828435 } ]</td>
</tr>
<tr>
<td>Image format</td>
<td>&quot;Jpeg&quot;</td>
</tr>
<tr>
<td>Image dimensions</td>
<td>1080 x 1620</td>
</tr>
</tbody>
</table>

https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/
External services integration
Microsoft Cognitive Services

DEMO

Analyze Image
Optical Character Recognition
Handwritten Text Recognition
Emotion Detection
Agenda

› Computer Vision
› Image Recognition
› Pepper
   › Softbank Robotics
   › Characteristics
   › Usecases
   › Computer Vision
   › External services
› On-device CV with CNNs
On-device CV with CNNs

Why

› Privacy
› Latency
› Connectivity
› Security
› Cost
On-device CV with CNNs

Limitations

› Compute
› Memory
› Storage
› Power
› Bandwidth
On-device CV with CNNs

How

› e.g. Tensorflow Mobile
› or Tensorflow Lite

Pre-trained models
Tensorflow Object Detection API
...Out of curiosity

Google’s Algorithm found houses

Google’s Algorithm found no houses
Yes, but: chihuahua or muffin?
Comparison

› **Amazon’s Rekognition** is not just good at identifying the primary object but also the many objects around it

› **Google’s Vision API and IBM Watson Vision** return straightforward, descriptive labels

› **Microsoft’s** tags were usually too high level

› **Cloudsight** is a hybrid between human tagging and machine labelling. More accurate. Slower. More expensive.

› **Clarifai** returns, by far, the most tags (at 20) although very generic tags. It also adds qualitative and subjective labels, such as “cute”, “funny”, “adorable”, and “delicious”