How good is AI at facial emotion recognition?

February 16, 2024

Plan of the presentation

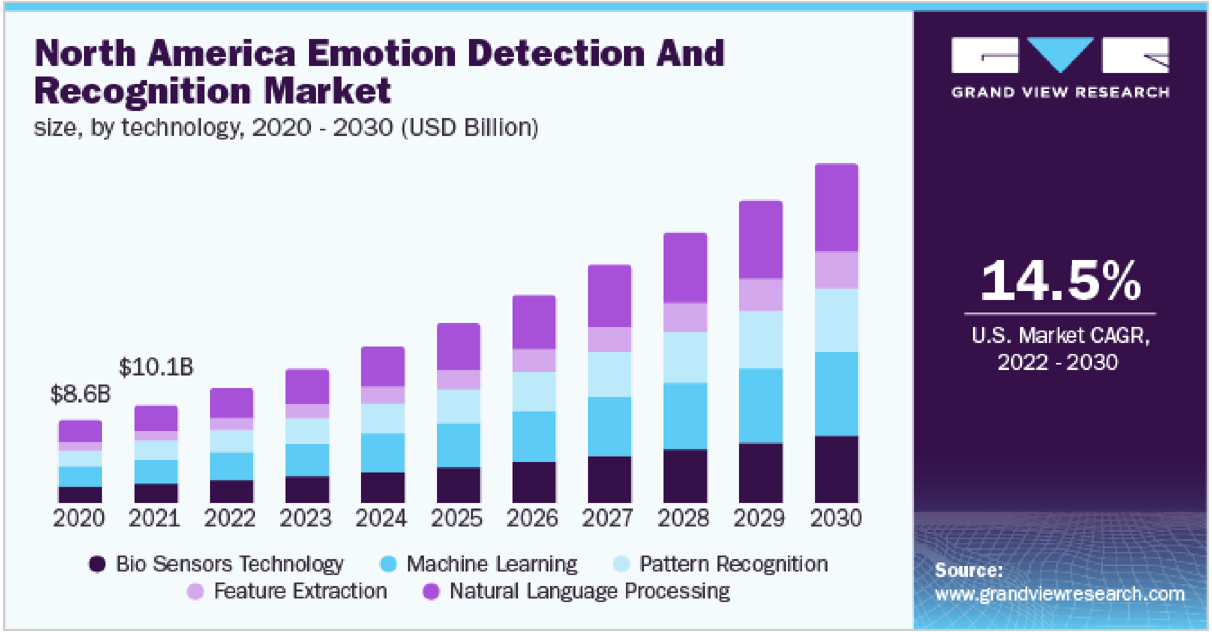
- **Introduction and goals of the project**
- **Data collection and cleaning**
  - Data sources
  - Data cleaning
  - Creating SQL database
  - Exposing data with API
- **Exploratory Data analysis (EDA)**
  - Emotions in everyday life
- **Used models**
  - Deep Learning
  - Convolution Neural Networks (CNN)
  - Pre-trained CNN model
- **Summary**

Introduction

Emotions are what make us human.

Emotion recognition software uses artificial intelligence (AI) and machine learning to interpret human emotions from text, voice, facial expressions, and other non-verbal cues.

The global Emotion AI market generated **$1.8 billion in 2022** and is projected to reach **$13.8 billion by 2032.**



https://www.grandviewresearch.com/industry-analysis/emotion-detection-recognition-market-report

Introduction

Facial Emotion recognition

Entertainment	Healthcare	Automotive	Marketing	On-line meetings
• Face filters • Virtual make up • Emotion-aware Gaming	• Telemedicine • Patient Monitoring • Autism Spectrum Therapy	• Real-time driver monitoring • In-car personalization	• Test emotions in ads for targeted marketing • Evaluate user experience with digital interfaces	• Track people engagement • Analyse emotions during hiring interviews

GOALS

How good is AI facial emotion recognition?

What are used models? How do they work?

What factors affect the model? How does the quality and quantity of data impact the results?

What emotions can be detected?

Data Sources

- Kaggle (flat file)
- FER-2013 (flat file)
- API to gather more images
- Emotions in everyday life (csvm file)
- Web scrapping

Data cleaning

CSV file

Emotions in Everyday Life, D. Trampe, J. Quoidbach, M. Taguet,

10.1371/journal.pone.0145450

id	Hours	Day	Pride	Love	Hope	Gratitude	Joy	Satisfaction	Awe	...	Alertness	Anxiety	Disdain	Ofense	Guilt	Disgust	Fear	Embarassment	Sadness	Anger
1	1.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	14.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	14.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
1	14.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
1	15.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

- Change data types to numeric
- Remove rows with missing values (1233 rows)

Shape

(**69544**, 21) → (**68311**, 21)

Number of unique values for „id”

12211 → 12108

Data cleaning

Web scrapping

facs_single_units

Action Unit	Description	Facial Muscle	Example
1	Inner Brow Raiser	Frontalis, pars medialis	https://imotions.com/wp-content/uploads/2022/1...
2	Outer Brow Raiser (unilateral, right side)	Frontalis, pars lateralis	https://imotions.com/wp-content/uploads/2022/1...
4	Brow Lowerer	Depressor Glabellae, Depressor Supercilli, Cur...	https://imotions.com/wp-content/uploads/2022/1...
5	Upper Lid Raiser	Levator palpebrae superioris	https://imotions.com/wp-content/uploads/2022/1...

facs_emotions_units

Emotion	Action Units	Description
Happiness / Joy	6 + 12	Cheek Raiser, Lip Corner Puller
Sadness	1 + 4 + 15	Inner Brow Raiser, Brow Lowerer, Lip Corner De...
Surprise	1 + 2 + 5 + 26	Inner Brow Raiser, Outer Brow Raiser, Upper Li...
Fear	1 + 2 + 4 + 5 + 7 + 20 + 26	Inner Brow Raiser, Outer Brow Raiser, Brow Low...
Anger	4 + 5 + 7 + 23	Brow Lowerer, Upper Lid Raiser, Lid Tightener,...
Disgust	9 + 15 + 16	Nose Wrinkler, Lip Corner Depressor, Lower Lip...
Contempt	12 + 14 (on one side of the face)	Lip Corner Puller, Dimpler

- Cleaning values in column *Action Units*
- Exploding the table with *emotion units* to connect with table containing *single units*

Data cleaning

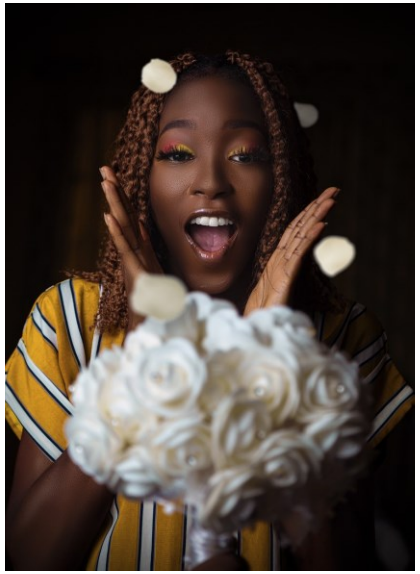
API images

Removing some picturess, or movimg them to different categories

„Anger”

„Fear”

Angry -> surprised



Data cleaning

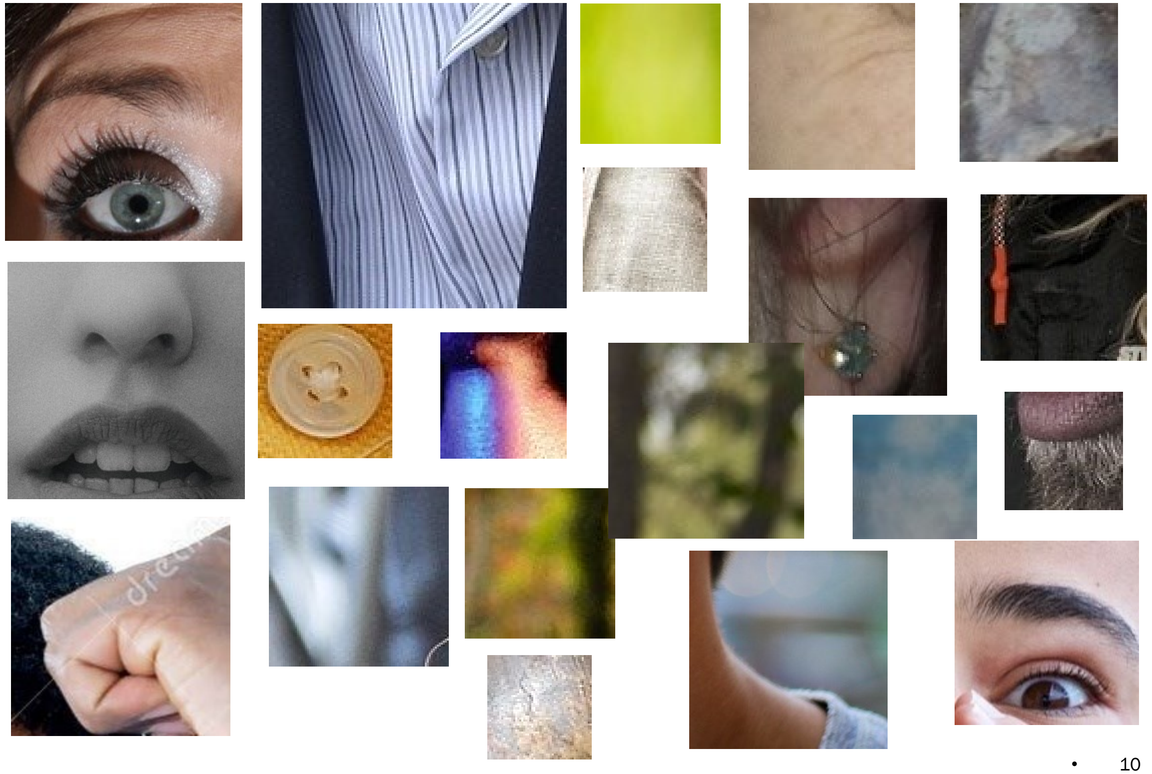
API images

Haarcascade classifier from OpenCV for face detection

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

Face not detected

Detected as „face”



Data cleaning

API images

PixaBay	Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise	Total
Initial	47	7	56	50	50	50	51	311
First selection	26	5	7	36	33	17	23	147
Face detection	**18**	**3**	**4**	**26**	**25**	**15**	**18**	**109**

DuckDuckGo	Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise	Total
Initial	84	99	95	97	92	93	94	654
First selection	73	71	70	89	59	71	88	521
Face detection	62	63	55	79	58	51	80	**448**

Data cleaning

Images from flat files(data base)

	Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise	Total
Kaggle	890	439	570	1406	524	746	775	**5350**
FER-2013	3995	436	4097	7215	4965	4830	3171	**28709**

- Not balanced data, Disgusted' face is in minority
- Fer-2013 almost 6 time larger than Kaggle data set
- Randomly picked 200 images from each category

Data cleaning

Images into Data Frames

```
emotions = ['Anger', 'Disgust', 'Fear', 'Happiness', 'Neutral', 'Sadness', 'Surprise']
```

```
pixabay_path = '1_pixabay_images/1_Face_Extraction/'
```

```
pixabay = images_to_tables(pixabay_path, categories=emotions, source=1)
```

```
pixabay
```

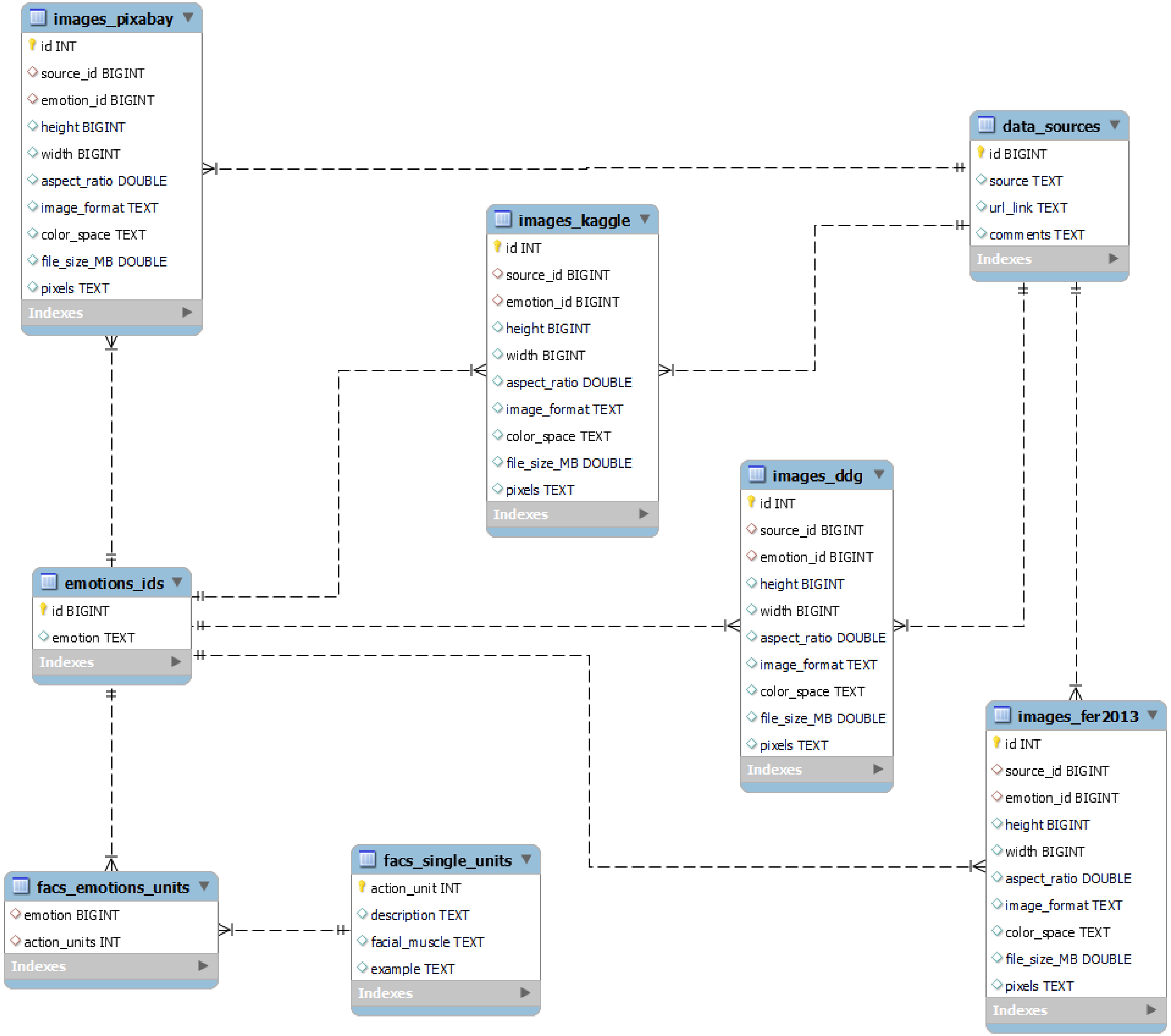
	source_id	emotion_id	height	width	aspect_ratio	image_format	color_space	file_size_MB	pixels
0	1	0	302	302	1.000000	JPG	RGB	0.022323	[[[255, 255, 255], [255, 255, 255], [255, 255,...
1	1	0	1218	1280	1.050903	JPG	RGB	0.439991	[[[68, 68, 68], [73, 73, 73], [63, 63, 63], [8...
2	1	0	620	620	1.000000	JPG	RGB	0.136570	[[[231, 231, 231], [232, 232, 232], [233, 233,...
3	1	0	410	410	1.000000	JPG	RGB	0.041554	[[[36, 36, 36], [36, 36, 36], [36, 36, 36], [3...
4	1	0	320	320	1.000000	JPG	RGB	0.078756	[[[255, 255, 255], [126, 126, 126], [79, 79, 7...
...	...	...	...	...	...	...	...	...	...

SQL DATABASE

List of tables

- all_images
- data_sources
- emotions_ids
- facs_emotions_units
- facs_single_units
- images_ddg
- images_fer2013
- images_kaggle
- images_pixabay

Total nr of images
3357



Exosing data with API

Welcome to the facial emotion recognition database

The available data was collected for facial emotion recognition project and includes images that capture different facial expressions. There are 3357 images in total and they originate from diverse data sources. For those downloaded via API, faces were initially detected and extracted from images. Images from Kaggle and FER-2013 already possess the appropriate format.

The available options for exploration are below. Please enter the selected option in the explore window.

To adjust the number of results per page, modify the associated **page** and **page_size** parameters:

`http://127.0.0.1:8080/images/`**`?page=1&page_size=50`**

To see list of all collected images use this link:

List of all images

To access a single image, go to /**images**/**{img_id}** and replace **{img_id}** with the image number (1-3357).

Explore Images based on the source:

1 : data from PixaBay API

2 : data from DuckDuckGo API

3 : data from Kaggle

4 : data from FER-2013

Source ID: [ ] Explore

Explore Images based on color:

0 : Grayscale

1 : RGB

Color: [ ] Exploret


```
// 20240213164729
// http://127.0.0.1:8080/images/emotions?emotion=5

{
  "images": [
    {
      "aspect_ratio": 1.0017921146953406,
      "color_space": "RGB",
      "emotion": "Surprise",
      "file_size_MB": 0.06939506530761719,
      "height": 558,
      "image_format": "JPG",
      "pixels": "[[[  1   2   6]\n  [  1   2   6]\n  [  1   2   6]\n  ...\n  [  4   5   9]\n  [  4   5   9]\n  [  4   5   9]]\n\n [[  1   2   6]\n  [  1   2   6]\n  [  1   2   6]\n  ...\n  [  4   5   9]\n  [  4   5   9]\n  [  4   5   9]]\n\n [[  1   2   6]\n  [  1   2   6]\n  [  1   2   6]\n  ...\n  [  4   5   9]\n  [  4   5   9]\n  [  4   5   9]]\n\n ...\n\n [[ 20 118 172]\n  [ 25 125 179]\n  [ 33 133 191]\n  ...\n  [  7  13  20]\n  [  7  13  20]\n  [  5  13  20]]\n\n [[ 16 116 170]\n  [ 21 121 175]\n  [ 26 126 184]\n  ...\n  [  7  13  20]\n  [  8  14  21]\n  [  6  14  21]]\n\n [[ 18 118 172]\n  [ 23 123 177]\n  [ 23 123 181]\n  ...\n  [  8  14  21]\n  [  8  14  21]\n  [  7  15  22]]]",
      "source": "Pixabay API",
      "url_link": "https://pixabay.com/service/about/api/",
      "width": 559
    },
```

Exploratory Data Analysis (EDA)

Emotions in every day life

id	Hours	Day	Pride	Love	Hope	Gratitude	Joy	Satisfaction	Awe	...
1	1.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1	14.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Year of data collection (February 2013 to April 2014)

- completed a total number of 65,721 emotion reports over an average of 35 days
- smartphone application monitored real-time emotions of an exceptionally large (N = 11,000+) and heterogeneous participants sample.

75% Female 25% Male

Age: 14-74 years

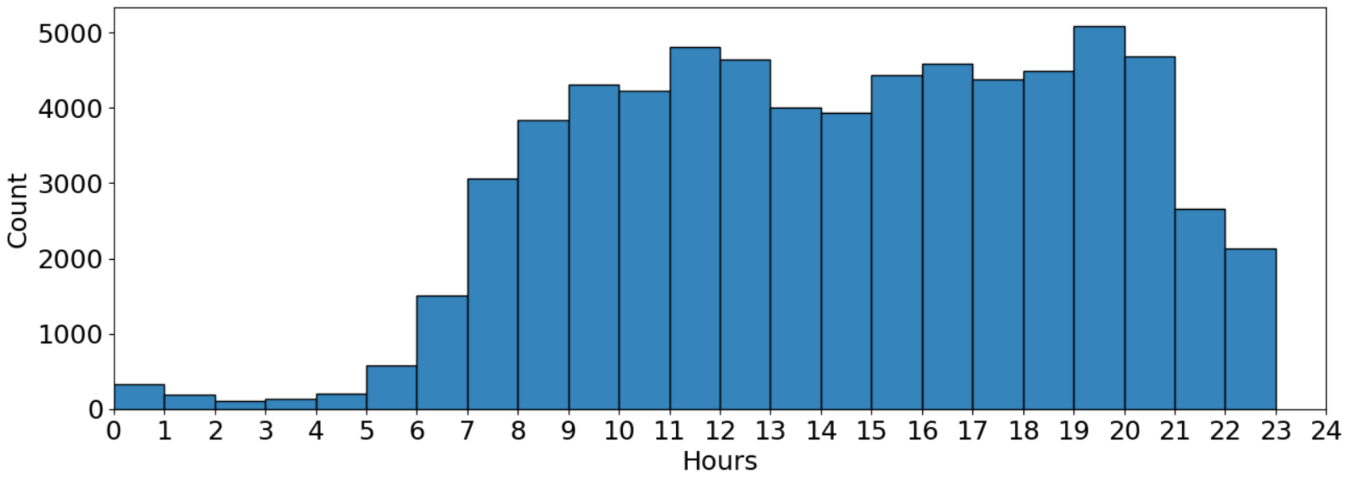
Nr of resonses : 1 - 263 per person

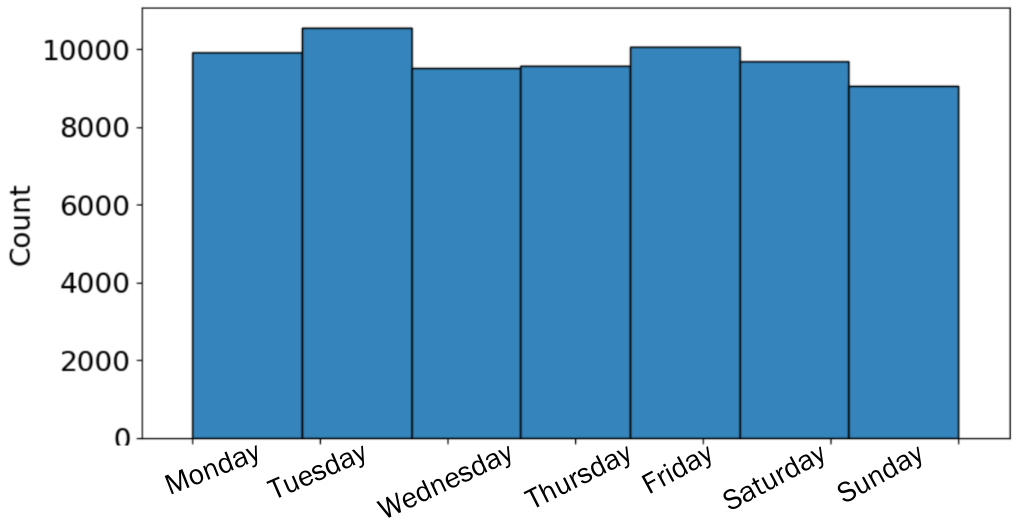
93% French, rest Swiss Belgian or other

- how often do people experience emotions in general?
- which emotions do people specifically experience?

Emotions in everyday life

Number of questioners completed during different days of the week and times of the day



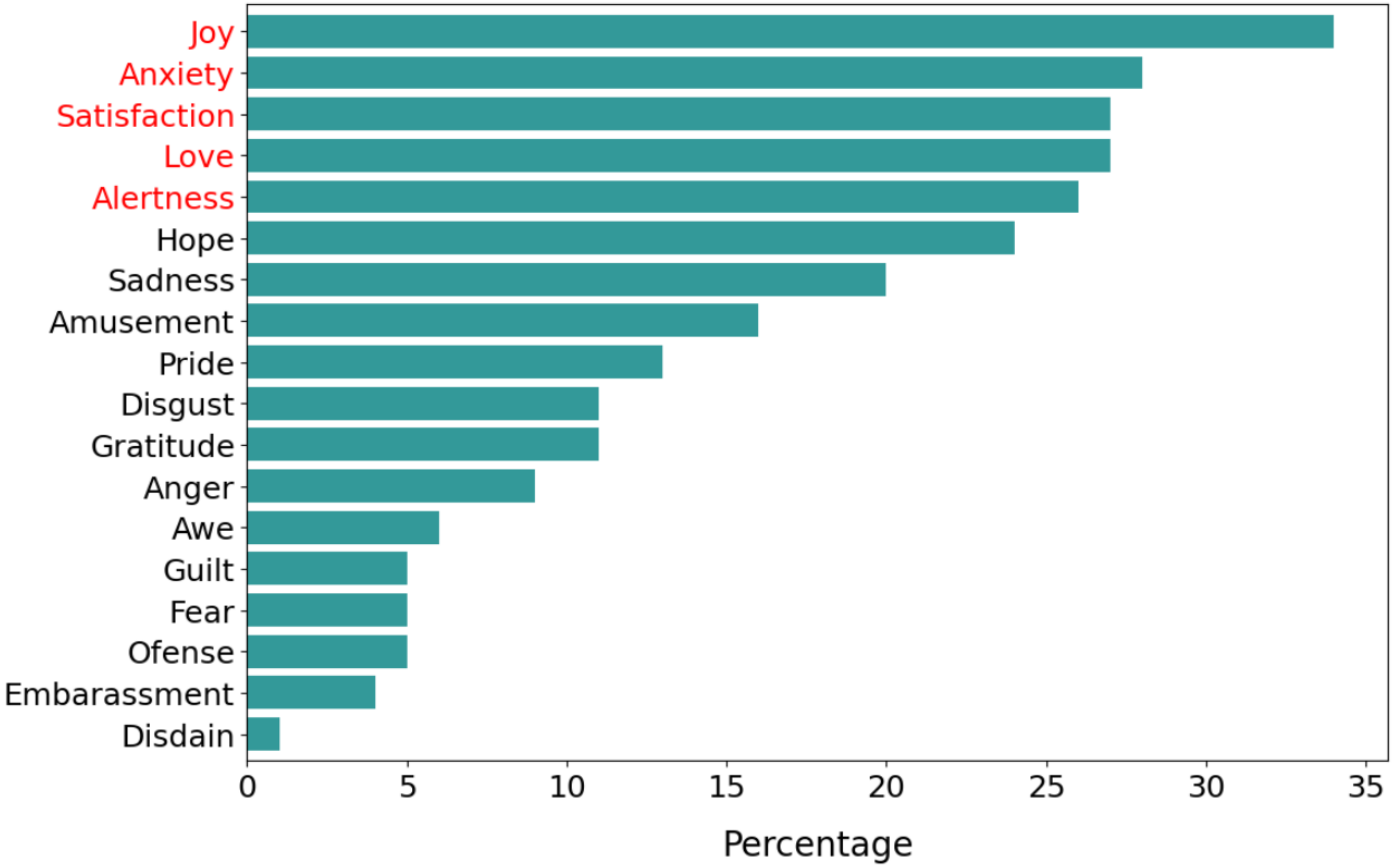


1614 only between 23-5

Excluded from further analysis

Evenly distributed across the week

Emotions in everyday life

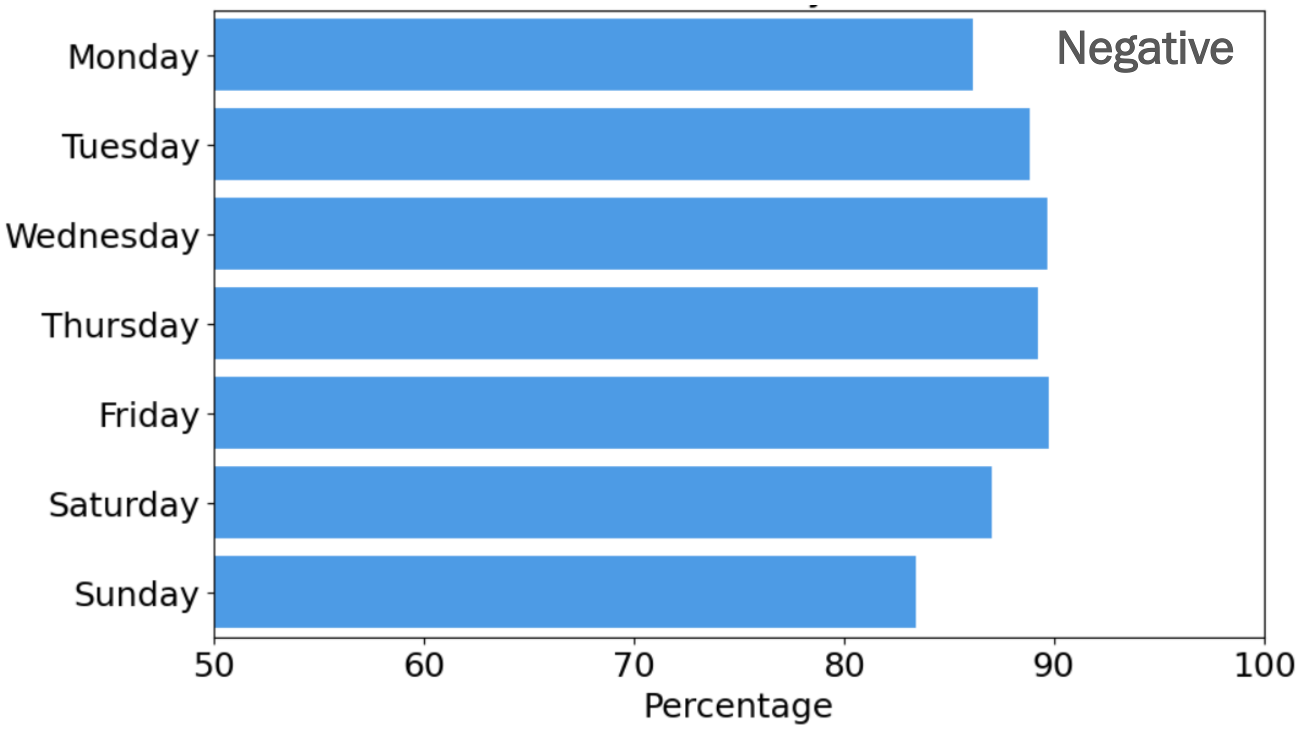


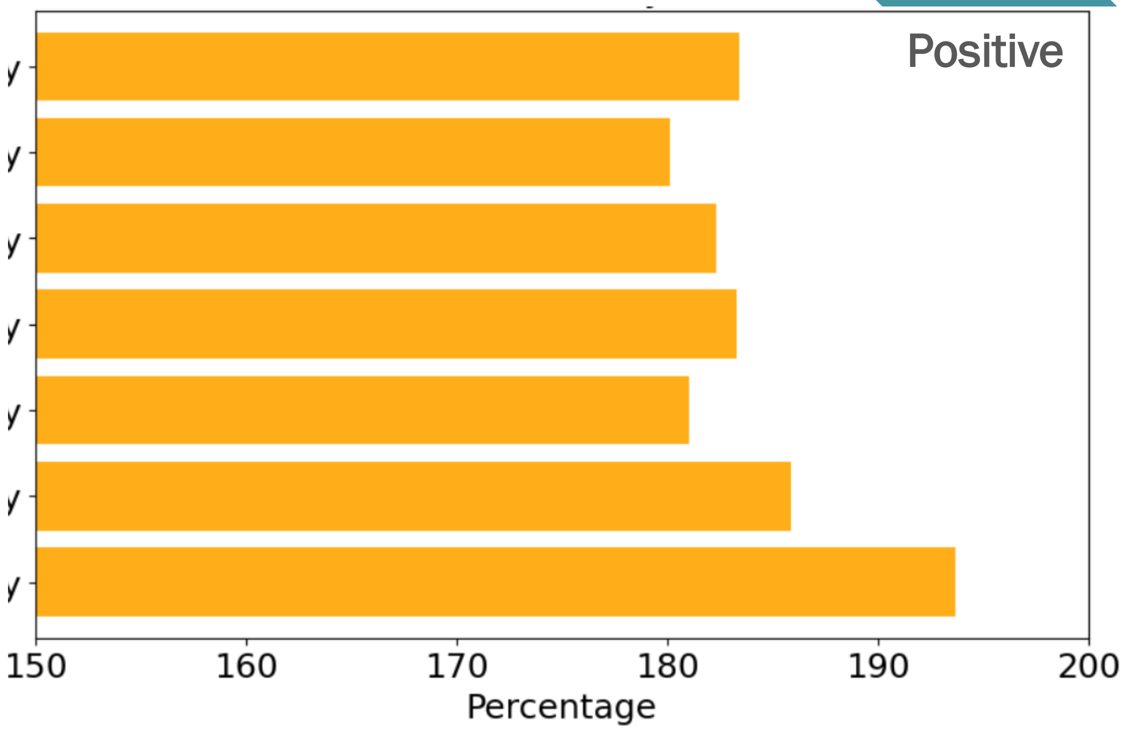
Positive	Negative
Joy	Anxiety
Satisfaction	Sadness
Love	Disgust
Alertness	Anger
Hope	Guilt
Amusement	Fear
Pride	Ofense
Gratitude	Embarrassment
Awe	Disdain

- Participants experienced **at least one emotion 90% of the time**.
- People experienced **positive emotions 2.5 times more often** than negative emotions.

Emotions in everyday life

How do emotions change throughout **the week**?



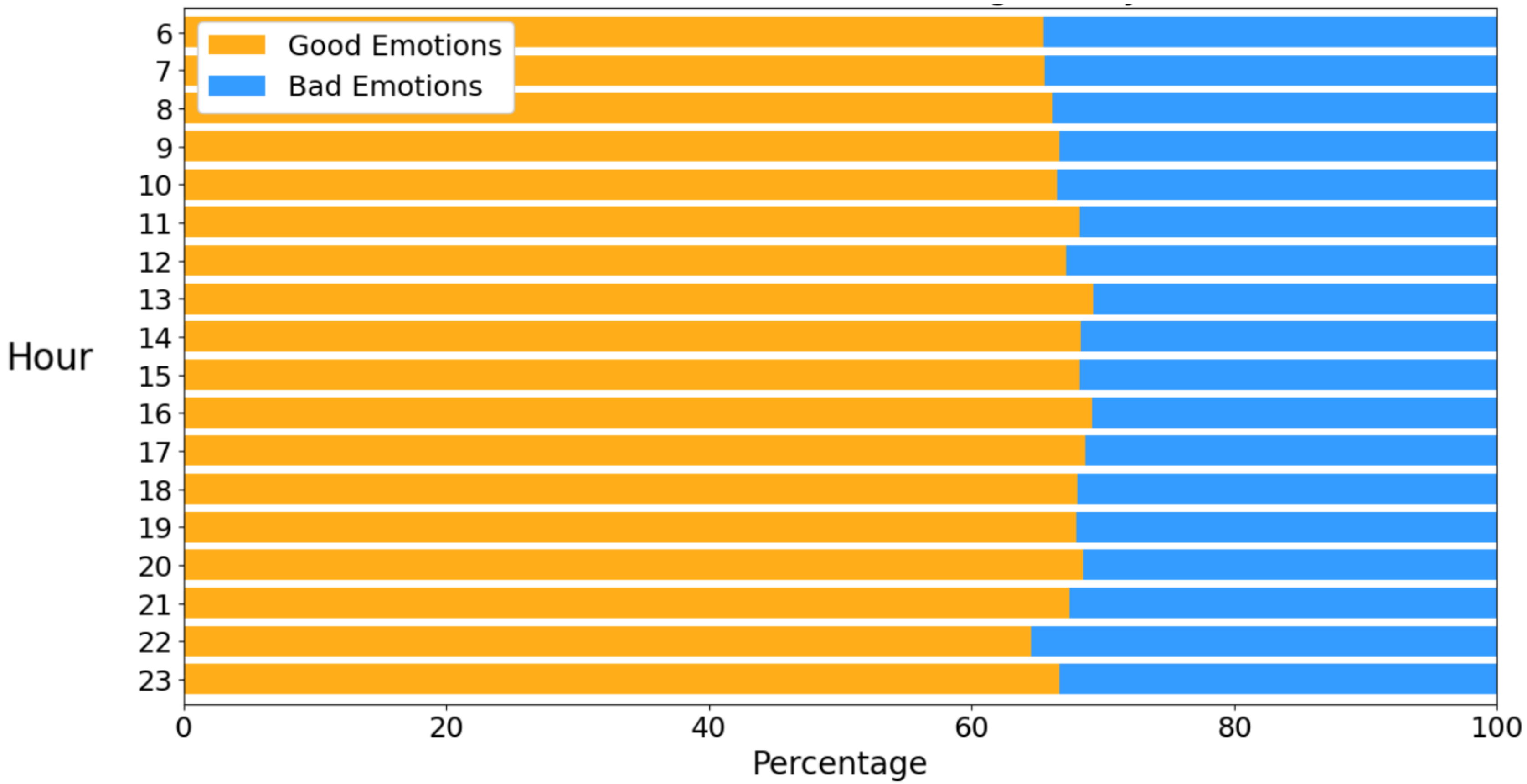


☹ We feel worse in the middle of the week

☺ We feel better during the weekend

Emotions in everyday life

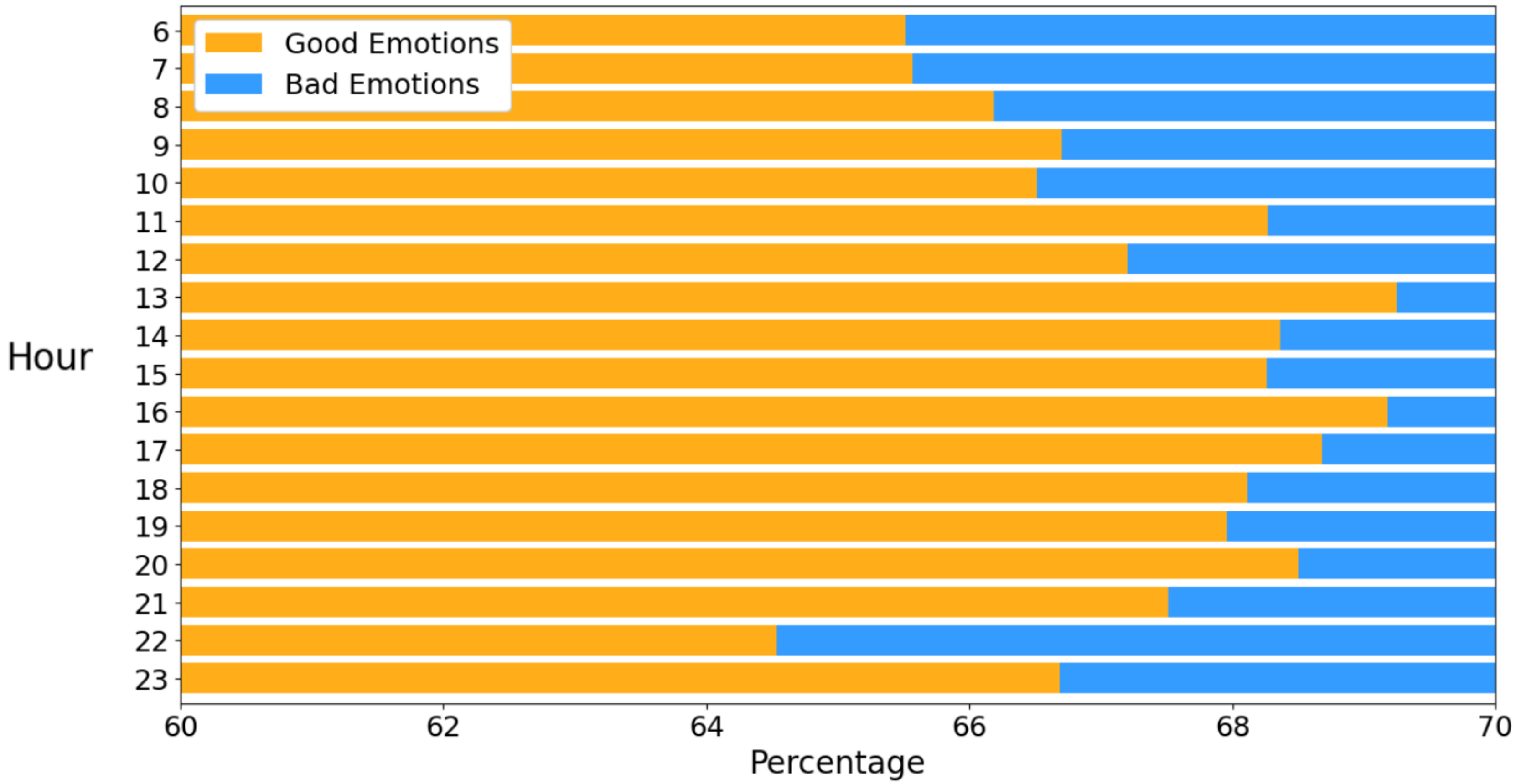
How do emotions change throughout **the day**?



- **Positive emotions dominate** – over 60% of time
- It seems that proportion of positive and negative emotions is not changing much during the day …

Emotions in everyday life

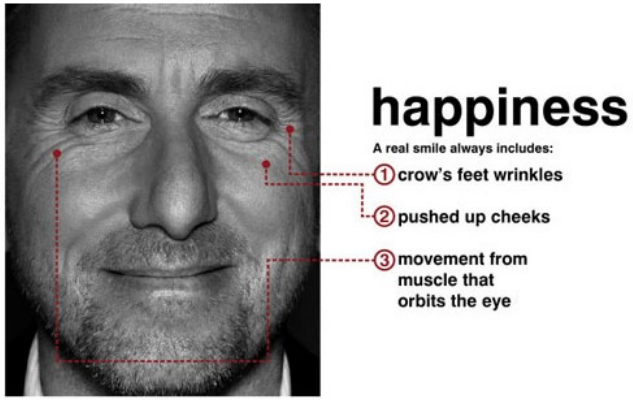
How do emotions change throughout **the day**?

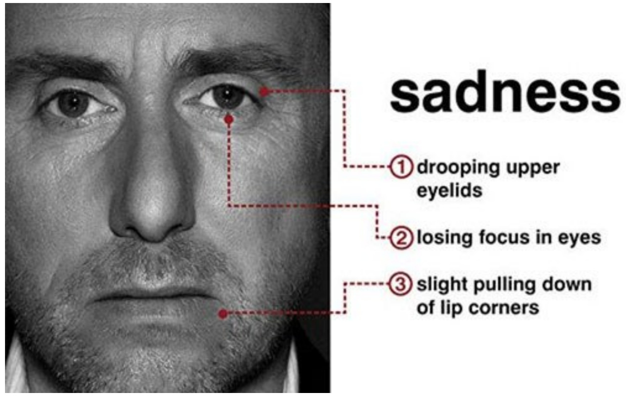


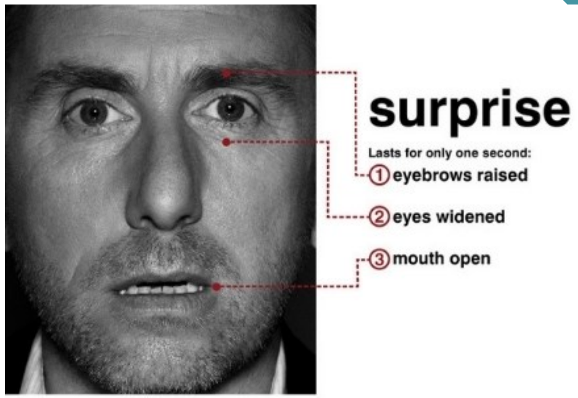
… but if we zoom in we can see fluctuations during the day

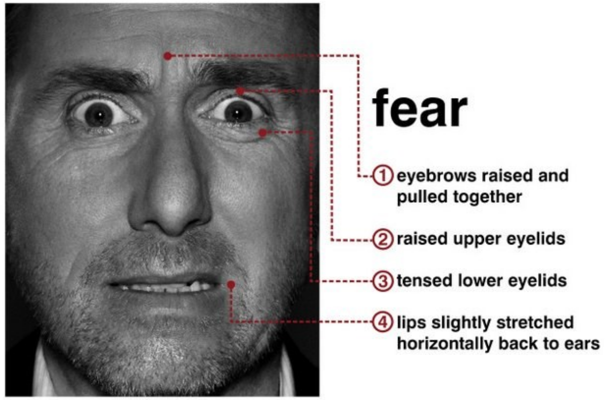
What emotions can be detected?

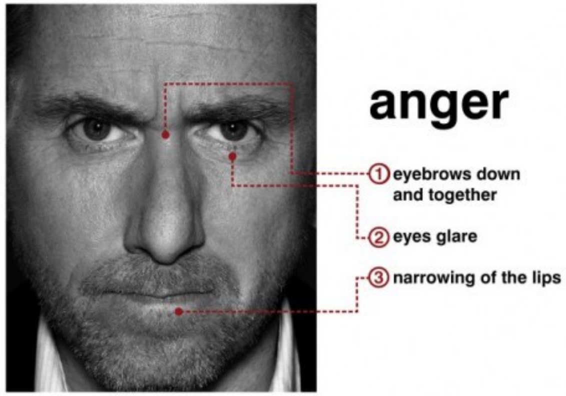
Universal facial expressions that we all use, even across cultural divides (research by Dr. Paul Ekman)

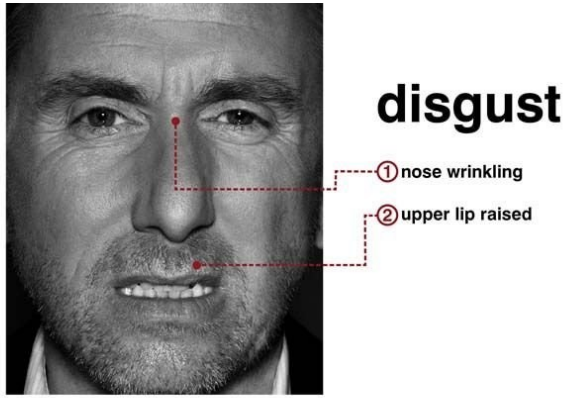








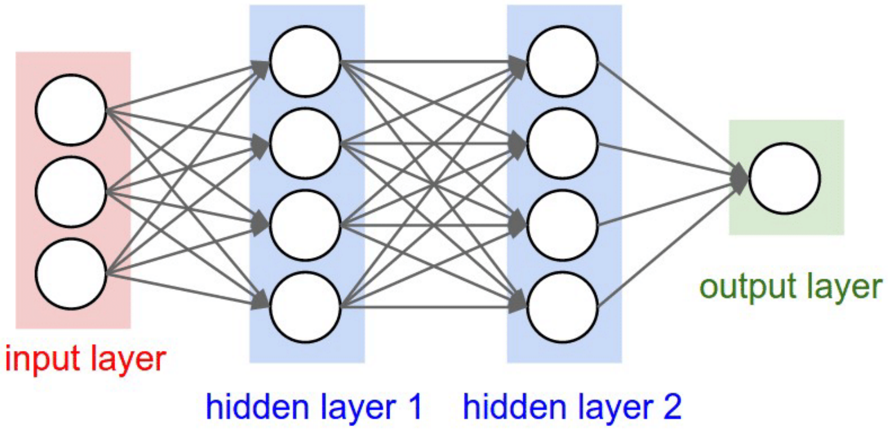




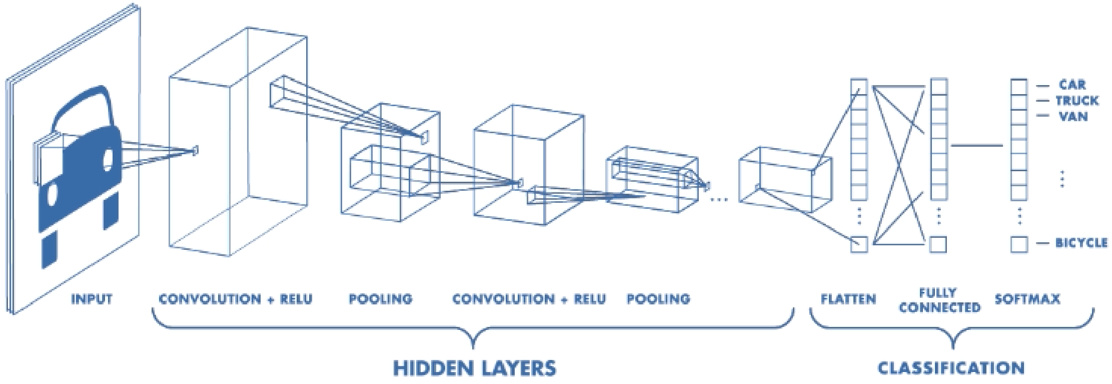
https://www.paulekman.com/

How?

Deep Learning

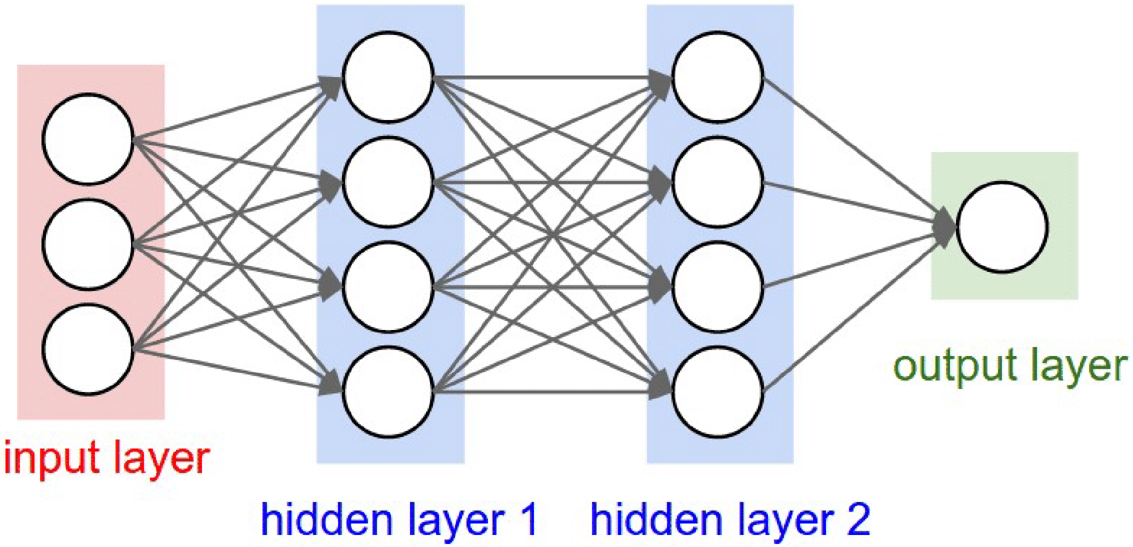


Convolutional Neural Networks



Pre-trained CNN model

Deep Learning




```
model = Sequential([
  Dense(2048, activation='relu', input_shape=(img_size*img_size,)),
  Dense(1024, activation='relu'),
  Dense(512, activation='relu'),
  Dense(256, activation='relu'),
  Dense(128, activation='relu'),
  Dense(7, activation='softmax'),
])

model.compile(
  optimizer=Adam(learning_rate=0.001),
  loss='categorical_crossentropy',
  metrics=['accuracy'],
)

history = model.fit(
  X_train,
  to_categorical(y_train),
  epochs=20,
  batch_size=60,
)
```

- Change all images to gray scale, put the same size
- Flatten the image before modeling
- Normalize values of pixels to be in range 0-1
- Split data into train and test set

Deep Learning

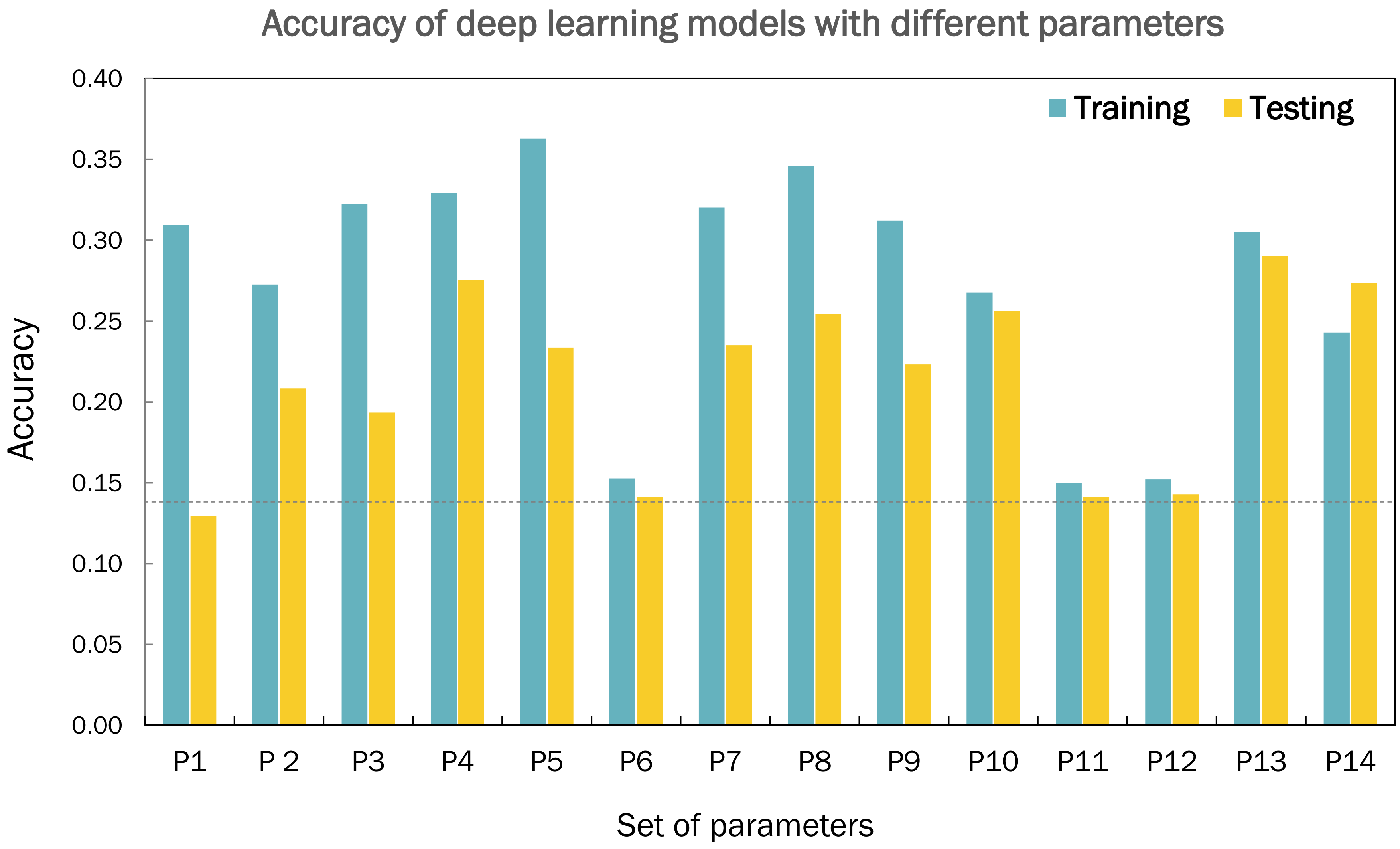
Testing of different parameters

Params.	Nr. layers	Nuerons/layer	Activation	Learning rate	Epochs	Batch_size	Training loss	Training accuracy	Time/epoch [s]	Total time [min]	Test loss	Test accuracy
P1	3	128/64/32	relu	0.001	10	50	1.79	0.31	6	1	1.95	0.13
P2	3	128/64/32	relu	0.001	10	60	1.86	0.27	5	1	2.21	0.21
P3	3	128/64/32	relu	0.001	15	60	1.76	0.32	5	1	2.09	0.19
P4	3	128/64/32	relu	0.001	20	60	1.74	0.33	5	2	1.89	0.28
P5	3	128/64/32	relu	0.001	30	60	1.68	0.36	5	3	1.95	0.23
P6	3	128/64/32	relu	0.01	20	60	1.95	0.15	5	2	1.95	0.14
P7	3	128/64/32	relu	0.0001	20	60	1.74	0.32	6	2	1.92	0.24
P8	3	256 / 64 / 32	relu	0.001	20	60	1.72	0.35	7	2	2.13	0.25
P9	4	256/128/64/32	relu	0.001	20	60	1.77	0.31	7	2	1.94	0.22
P10	5	256/128/64/32/16	relu	0.001	20	60	1.81	0.27	8	3	1.87	0.26
P11	6	512/128/64/32/16	relu	0.001	20	60	1.95	1.95	16	5	1.95	0.14
P12	6	512/128/64/32/16	sigmoid	0.001	20	60	1.95	0.15	14	5	1.95	0.14
P13	5	1024/512/256/128/64	relu	0.001	20	60	1.77	0.31	26	9	1.84	0.29
P14	5	2048/1024/512/256/128	relu	0.001	20	60	1.87	0.24	50	17	1.86	0.27

224x224 pixels

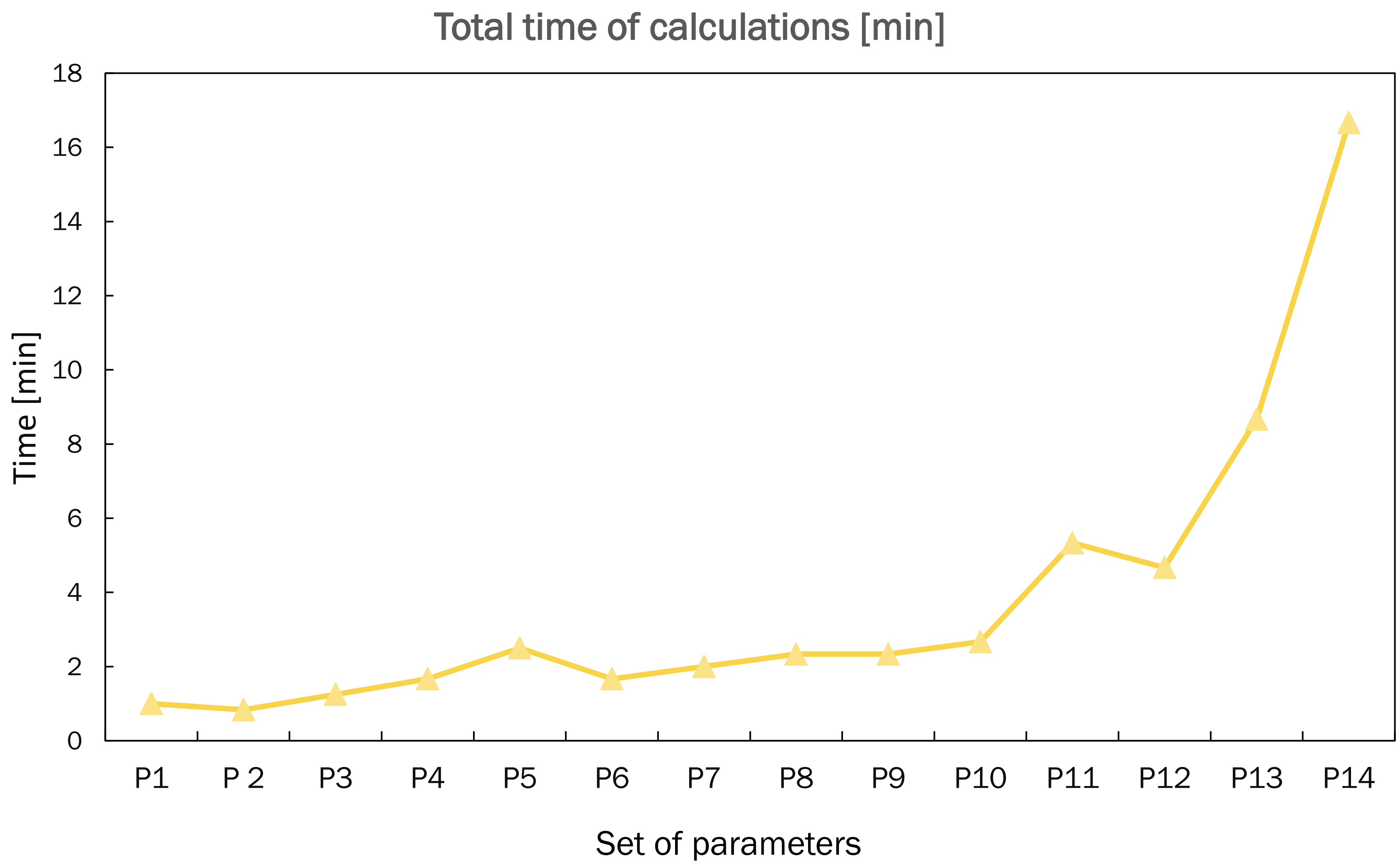
Deep Learning

224x224 pixels



Deep Learning

224x224 pixels



Deep Learning

64x64 pixels

Decrease the image size to speed up calculations?

Nr. layers	Nuerons/layer	Activation	Learning rate	Epochs	Batch_size	Training loss	Training accuracy	Time/epoch [s]	Total time [min]	Test loss	Testing accuracy
5	**2048/1024/512/256/128**	relu	0.001	20	60	1.80	0.26	6	6	1.86	0.25
6	**2048/1024/512/256/128/64**					1.77	0.28	6	6	1.88	0.24
6	4096/2048/1024/512/256/128					1.95	0.15	15	15	1.95	0.14
7	4096/2048/1024/512/256/128/64					1.85	0.22	15	15	1.88	0.20
3+damping	512/d0.5/256/d0.5/128/d0.5					1.95	0.15	2	2	1.95	0.14

Deep Learning

Decrease the image size to speed up calculations?

64x64 pixels

Nr. layers	Nuerons/layer	Activation	Learning rate	Epochs	Batch_size	Training loss	Training accuracy	Time/epoch [s]	Total time [min]	Test loss	Testing accuracy
5	**2048/1024/512/256/128**	relu	0.001	20	60	1.80	0.26	6	6	1.86	0.25
6	**2048/1024/512/256/128/64**					1.77	0.28	6	6	1.88	0.24
6	4096/2048/1024/512/256/128					1.95	0.15	15	15	1.95	0.14
7	4096/2048/1024/512/256/128/64					1.85	0.22	15	15	1.88	0.20
3+damping	512/d0.5/256/d0.5/128/d0.5					1.95	0.15	2	2	1.95	0.14
5	2048/1024/512/256/128	relu	0.001	20	60	1.87	0.24	50	17	1.86	0.27

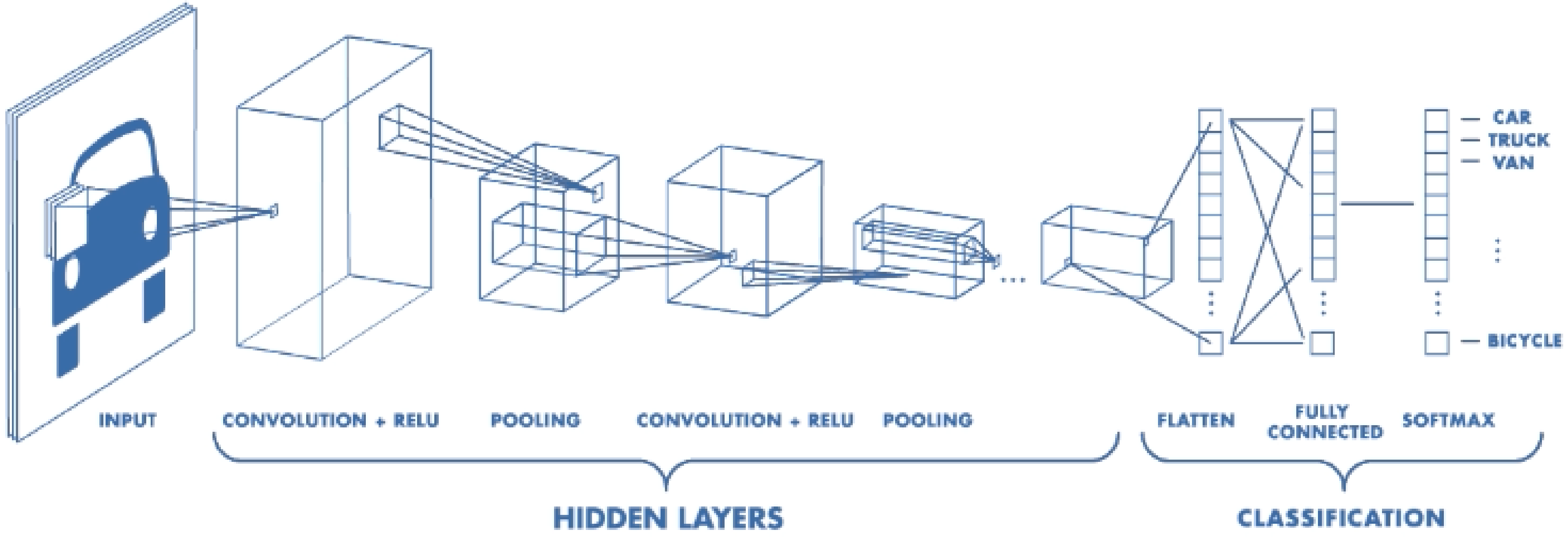
224x224 pixels

With smaller image size the **accuracy was similar**, and the **calculation were faster**

However, making the model more complex leads to a decrease in accuracy …

Convolutional Neural Networks

Specially designed to work with images



CNNs can retain spatial information as they take the images in the original format.

Convolutional Neural Networks

- **No need of flattening the images**
- Change all images to gray scale, resize
- Normalize values of pixels to be in range 0-1
- Split data into train and test set

Convolutional Neural Networks

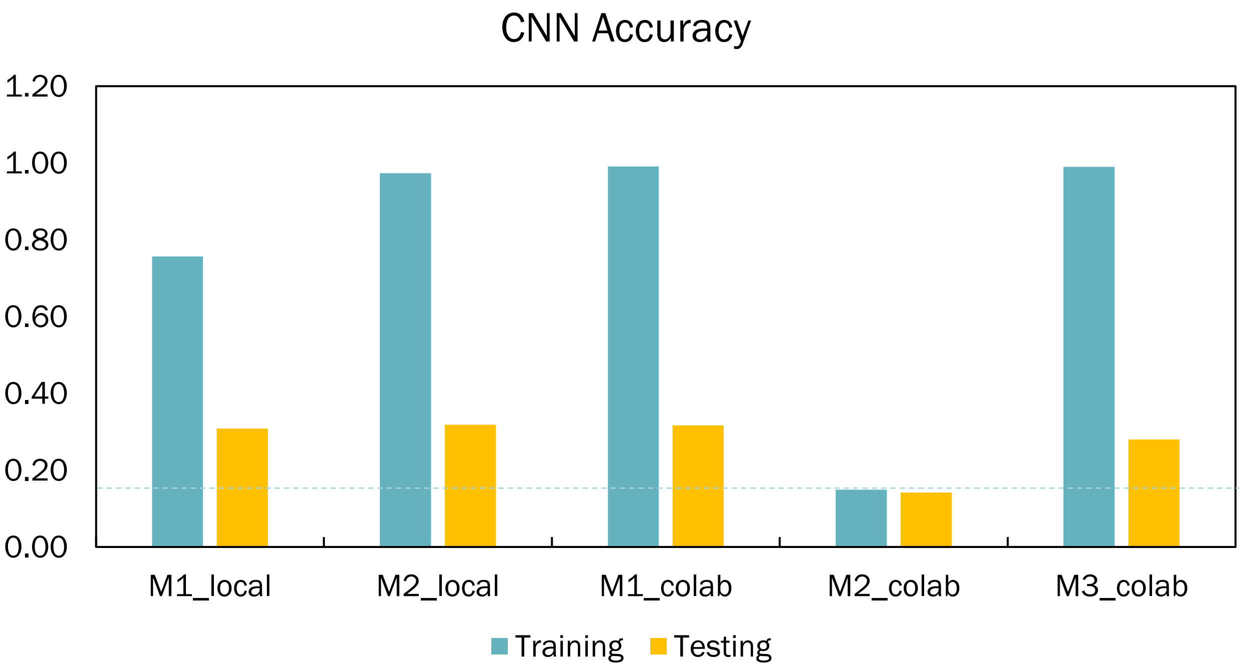
- Try different models

224x224 pixels

CNN	Descrition	Activation	Learning rate	Epochs	Batch_size	Training loss	Training accuracy	Time/epoch [s]	Total time [min]	Test loss	Testing accuracy
Model 1	Local computer	relu	0.001	5	60	0.75	0.76	60.00	5.00	2.15	0.31
Model 2	Local computer	relu	0.001	10	64	0.15	0.97	60.00	10.00	3.58	0.32
Model 1	colab	relu	0.001	30	64	0.04	0.99	18.00	9.00	4.64	0.32
Model 2	colab	relu	0.001	50	64	1.95	0.15	16.00	13.33	1.95	0.14
Model 3	colab	relu	0.001	50	32	0.03	0.99	33.00	27.50	3.79	0.28
Model 4	batch normalization	stopped 22 out of 50 epoch		50	32	0.05	0.99	43.00	15.77	3.30	0.30
Model 5	with padding	memory issues									
Model 6	data augmentation	memory issues, end of free limit on colab...									

Convolutional Neural Networks

- Overfitting?



224x224 pixels

CNN	Descrition	Activation	Learning rate	Epochs	Batch_size	Training loss	Training accuracy	Time/epoch [s]	Total time [min]	Test loss	Testing accuracy
Model 1	Local computer	relu	0.001	5	60	0.75	0.76	60.00	5.00	2.15	0.31
Model 2	Local computer	relu	0.001	10	64	0.15	0.97	60.00	10.00	3.58	0.32
Model 1	colab	relu	0.001	30	64	0.04	0.99	18.00	9.00	4.64	0.32
Model 2	colab	relu	0.001	50	64	1.95	0.15	16.00	13.33	1.95	0.14
Model 3	colab	relu	0.001	50	32	0.03	0.99	33.00	27.50	3.79	0.28
Model 4	batch normalization	stopped 22 out of 50 epoch		50	32	0.05	0.99	43.00	15.77	3.30	0.30
Model 5	with padding	memory issues									
Model 6	data augmentation	memory issues, end of free limit on colab...									

Pre-trained CNN model

Load pre-trained CNN model

```
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
```

Take input layer, and one of tha lest output layers

```
base_input = base_model.layers[0].input
base_output = base_model.layers[-3].output
```

Modify last layers to train on new data

```
final_output = GlobalAveragePooling2D()(base_output)
final_output = layers.Dense(128, activation='relu')(final_output)
final_output = layers.Dense(64, activation='relu')(final_output)
final_output = layers.Dense(7, activation='softmax')(final_output)

new_model = keras.Model(inputs=base_input, outputs=final_output)
```

- Pre-trained on color images, expected format (img_size, img_size, 3)
- Issues with image format....
- **Not so good results:**

  accuracy: 0.265

  time: 50 min

Summary

Models

- CNN little bit better than Deep Learning
- Pre-trained models did not help much
- The best accuracy up to 0.3
- Not so easy to tune parameters
- Problems with overfitting in CNN

What can be impoved?

- Increase data set to avoid overfitting or
- Use in built methods for data augmentation
- For emotion recognition - include facial features and not just compare full image
- Go beyond basic emotions.

Is AI good in recognizing human emotions?

http://localhost:8502/

36