

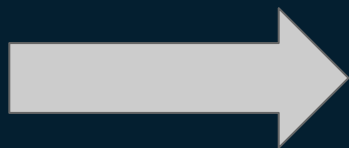
Age recognition

Big Data for Official Statistics
A.Y. 2020/2021

S. Palaia - Master Degree in Data Science - Sapienza

1. Objectives

Predict the person's age given a face picture



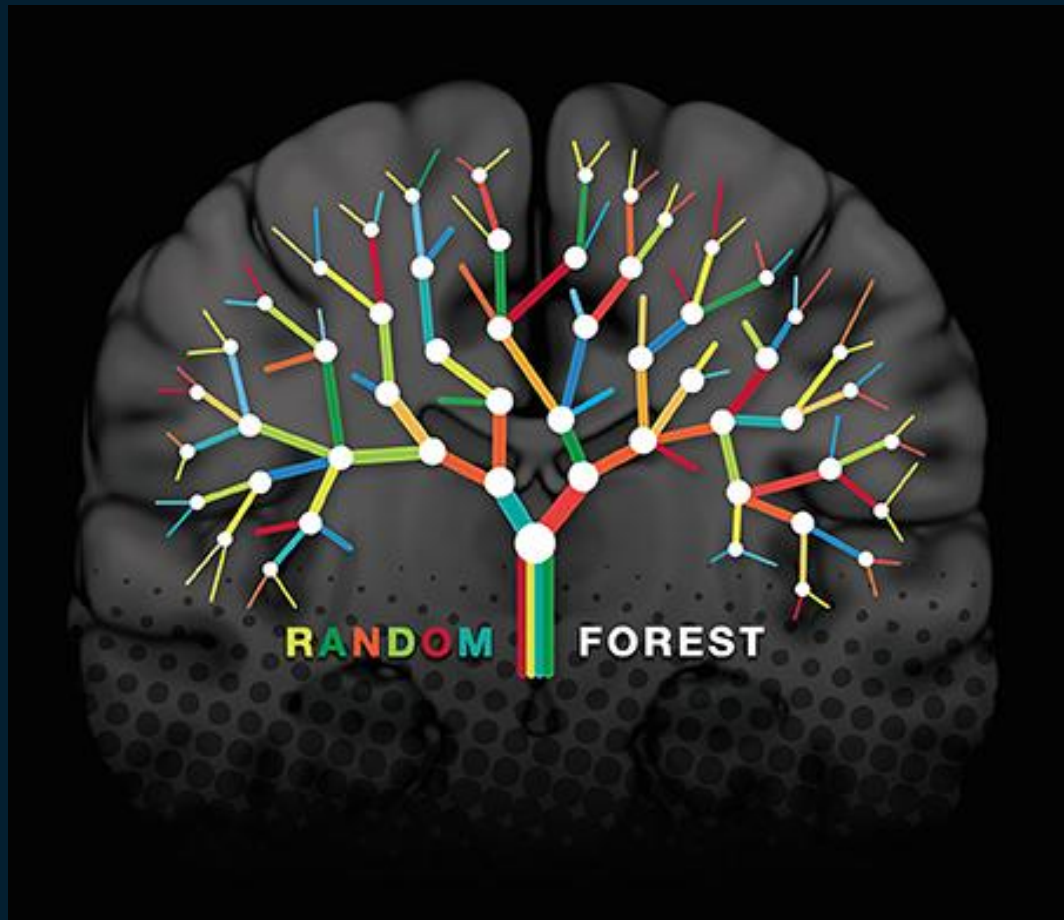
20-30?



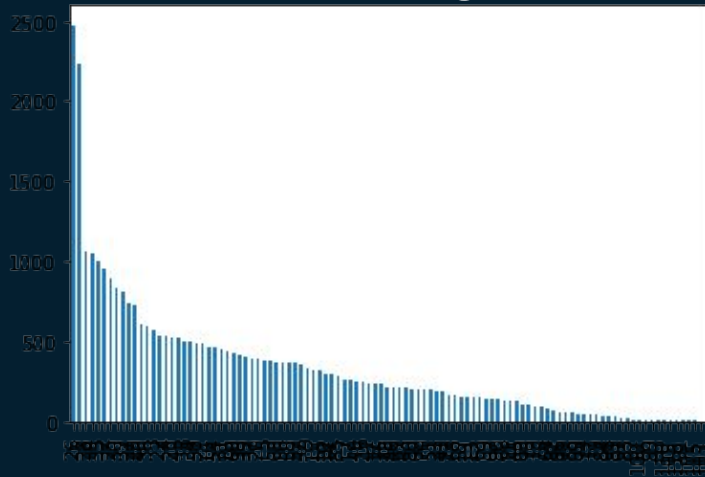
2. Dataset(s)

Name	Dimension of the Dataset (num. of sample)	Dimension of the pics (pixel width x pixel height)	Link
Facial-age (Kaggle)	9778	200px*200px	https://www.kaggle.com/frabbisw/facial-age
UTK Face	23.708	200px*200px	https://susanqq.github.io/UTKFace/
Union	33.486	200px*200px	

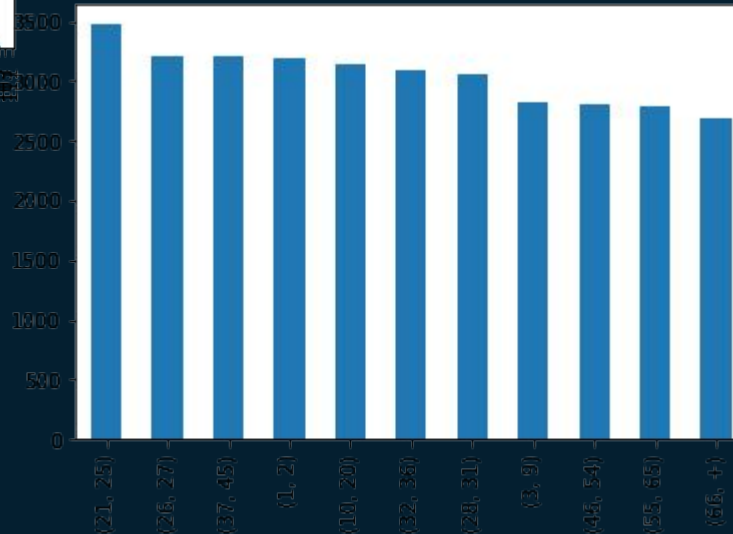
3. ML Project



3. Data cleaning and reshaping



Distribution of the ages/ages range
before and after preprocessing



3. Preprocessing - informations



```
1 faces.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 33486 entries, 0 to 33485  
Data columns (total 2 columns):  
#   Column  Non-Null Count  Dtype  
---  ---      -  
0    file    33486 non-null    object  
1    age     33486 non-null    object  
dtypes: object(2)  
memory usage: 523.3+ KB
```

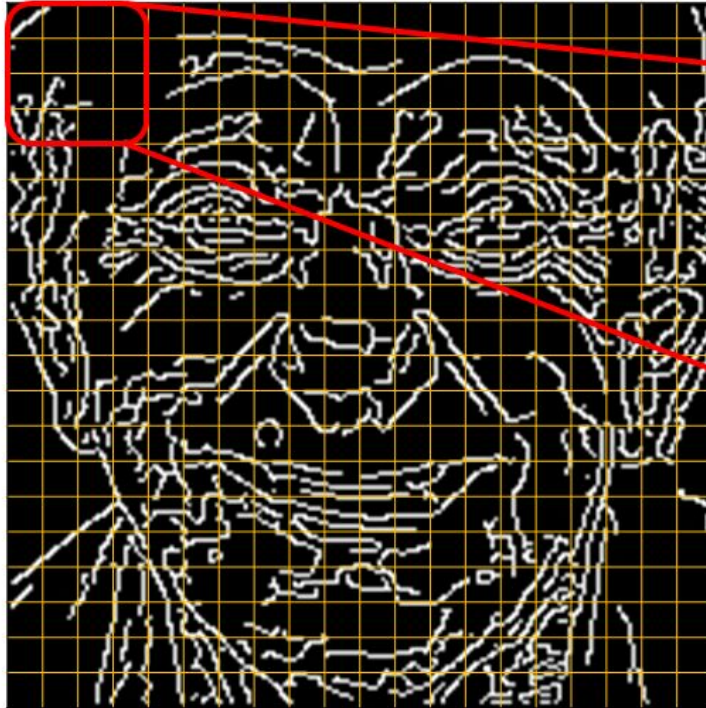
```
[ ] 1 faces=pd.read_csv(BASE_PATH + 'faces.csv')  
2 faces.head()
```

	file	age
0	48_65.jpg	(46, 54)
1	28_935.jpg	(28, 31)
2	4_294.jpg	(3, 9)
3	81_8.jpg	(66, +)
4	10_129.jpg	(10, 20)

```
[ ] 1 faces.head()
```

	file	age
0	48_65.jpg	8
1	28_935.jpg	4
2	4_294.jpg	5
3	81_8.jpg	10
4	10_129.jpg	1

3. Preprocessing - images



Sec 1	Sec 2	Sec 3	Sec 4
Sec 21	Sec 22	Sec 23	Sec 24
Sec 41	Sec 42	Sec 43	Sec 44
Sec 61	Sec 62	Sec 63	Sec 64

Mean and Standard Deviation

3. Preprocessing - images (cnt)



3. Preprocessing - images (cnt)

```
[ ] 1 features
```

	sec1_mean	sec1_std	sec2_mean	sec2_std	sec3_mean	sec3_std	sec4_mean	sec4_std
0	0.19	0.392301	0.00	0.000000	0.00	0.000000	0.10	0.300000
1	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000
2	0.08	0.271293	0.00	0.000000	0.23	0.420833	0.03	0.170587
3	0.10	0.300000	0.09	0.286182	0.00	0.000000	0.09	0.286182
4	0.19	0.392301	0.10	0.300000	0.00	0.000000	0.00	0.000000
...
66965	0.08	0.271293	0.18	0.384187	0.11	0.312890	0.03	0.170587
66966	0.00	0.000000	0.01	0.099499	0.19	0.392301	0.06	0.237487
66967	0.21	0.407308	0.07	0.255147	0.00	0.000000	0.00	0.000000
66968	0.12	0.324962	0.11	0.312890	0.05	0.217945	0.08	0.271293
66969	0.00	0.000000	0.10	0.300000	0.00	0.000000	0.08	0.271293

66970 rows × 801 columns

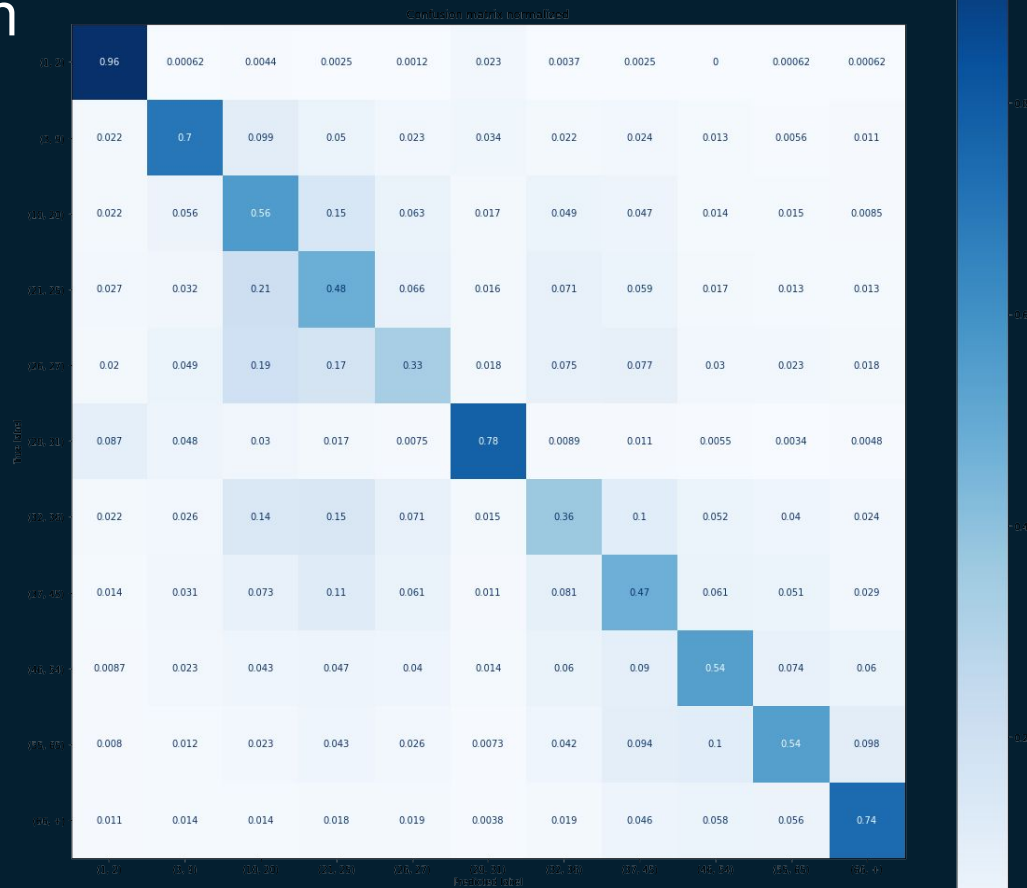
```
[ ] 1 features.to_csv(BASE_PATH+'/canny_features_extraction.csv', index=False)
```

4. Model(s)

- Logistic Regression
- Random Forest
- Stochastic Gradient Descent
- KNN
- Gaussian NB
- Perceptron
- Linear SVC
- Decision Tree
- SVC

5. Evaluation

Accuracy: 0.58902

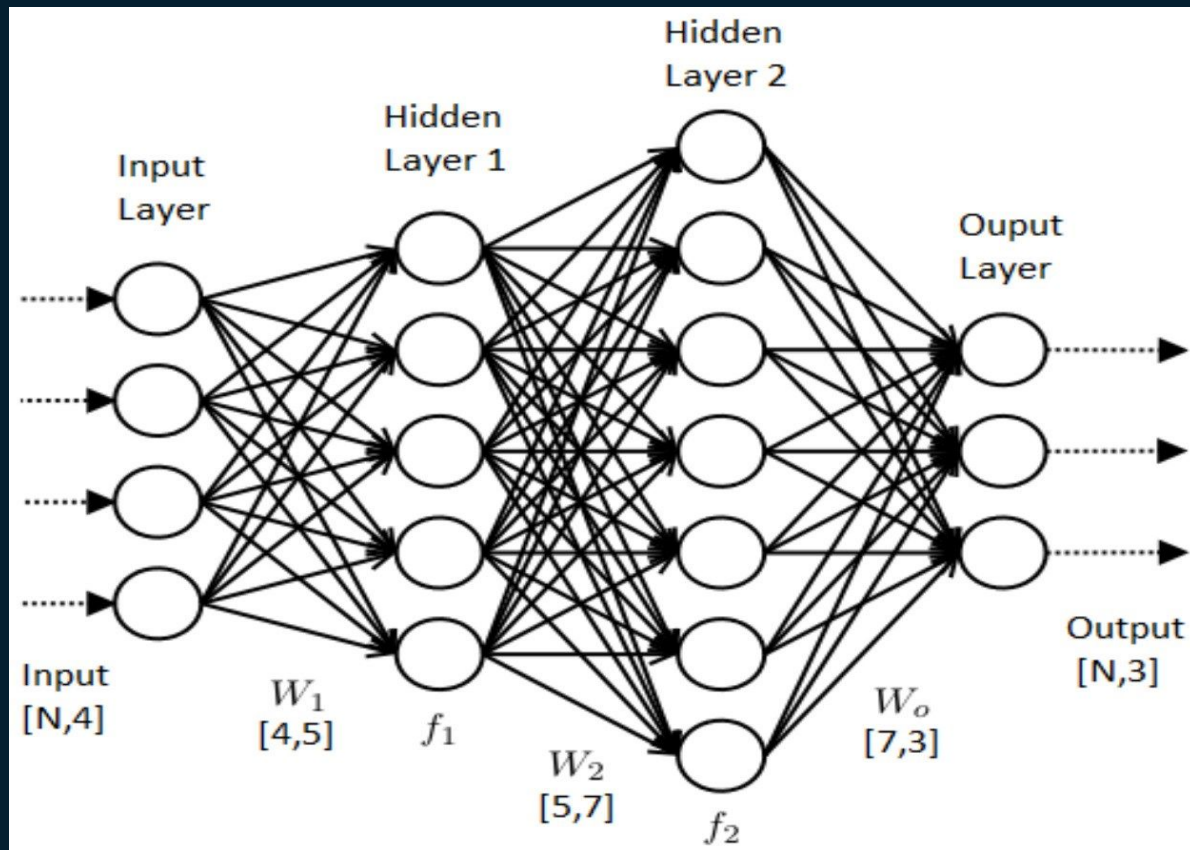


5. Evaluation (cnt)

```
[ ] 1 print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0.0	0.81	0.96	0.88	1604
1.0	0.71	0.70	0.71	1612
2.0	0.44	0.56	0.49	1761
3.0	0.39	0.48	0.43	1568
4.0	0.47	0.33	0.39	1579
5.0	0.82	0.78	0.80	1462
6.0	0.45	0.36	0.40	1514
7.0	0.48	0.47	0.48	1600
8.0	0.59	0.54	0.56	1372
9.0	0.64	0.54	0.59	1368
10.0	0.71	0.74	0.73	1303
accuracy			0.59	16743
macro avg	0.59	0.59	0.59	16743
weighted avg	0.59	0.59	0.58	16743

6. Final Project



3. Preprocess

- 7 ages range (encoded)
- GrayScale images
- Augmentation
- train/test/val -
21.486/6.000/6.000
- Batch 32
- Prefetch



4. Model - CNN (1)

4 kind of layers:

- Convolutional (2D)
- Average pooling
- Global Average pooling
- Dense

Number of filter increase, while dimension of pictures decrease

Relu inside
Softmax in the end

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 198, 198, 32)	320
average_pooling2d (AveragePo	(None, 99, 99, 32)	0
conv2d_1 (Conv2D)	(None, 97, 97, 64)	18496
average_pooling2d_1 (Average	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 46, 46, 128)	73856
average_pooling2d_2 (Average	(None, 23, 23, 128)	0
conv2d_3 (Conv2D)	(None, 21, 21, 256)	295168
average_pooling2d_3 (Average	(None, 10, 10, 256)	0
global_average_pooling2d (Gl	(None, 256)	0
dense (Dense)	(None, 132)	33924
dense_1 (Dense)	(None, 7)	931
Total params: 422,695		
Trainable params: 422,695		
Non-trainable params: 0		

5. Model - compile

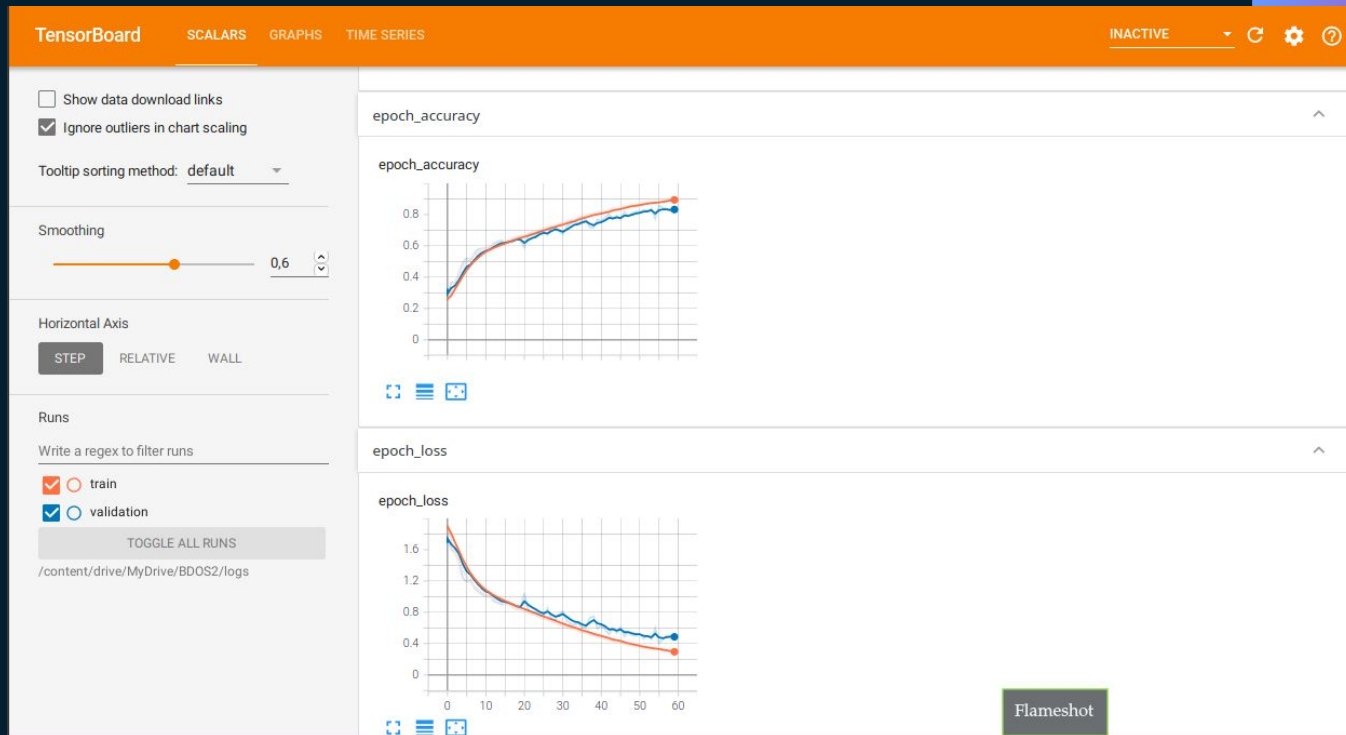
Sparse categorical cross entropy

Adam

Accuracy

Checkpoint

Tensor Board



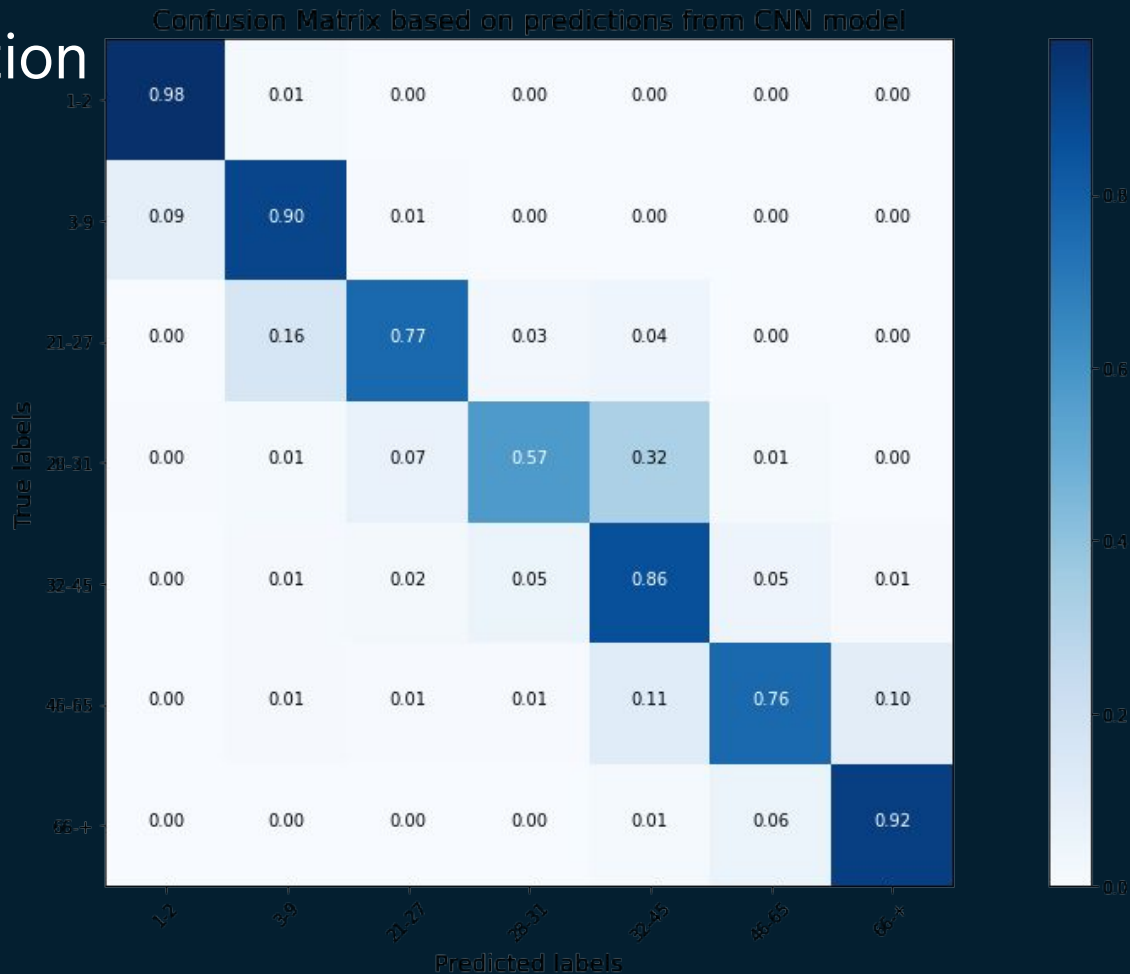
6. Evaluation

Accuracy:

Train - 0.88

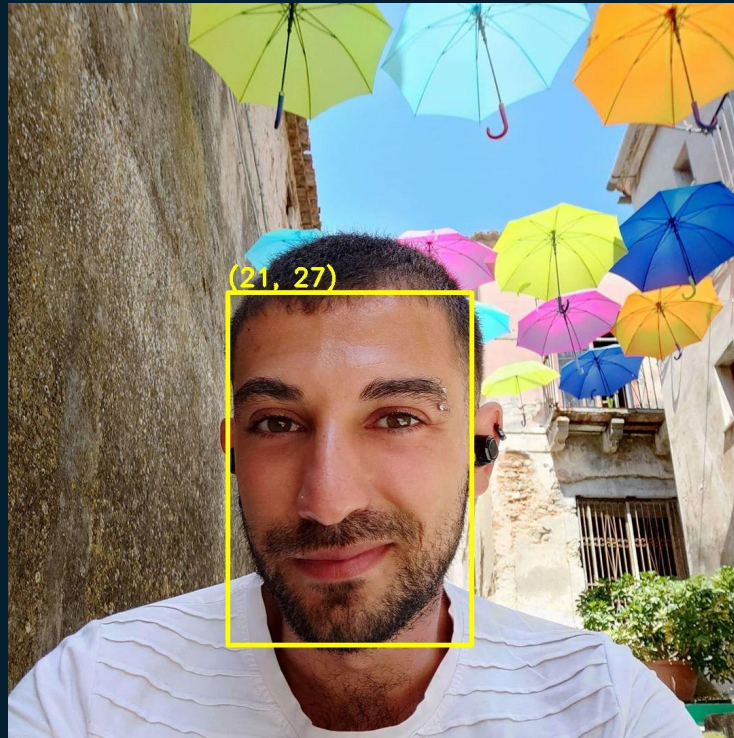
Valid - 0.86

Test - 0.84



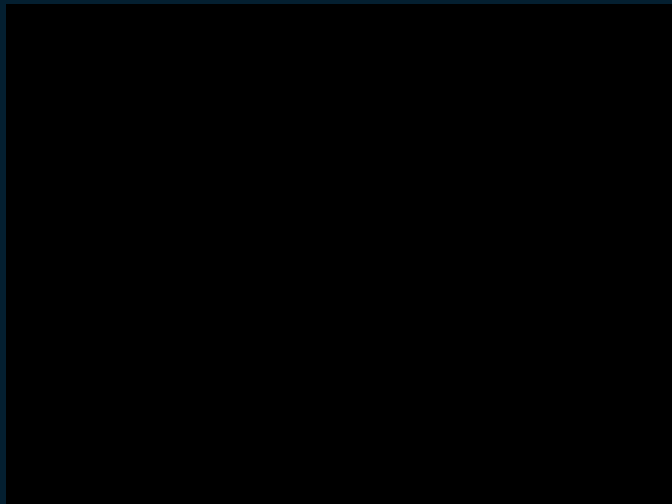
7. App

- Heroku - repository
- Flask
 - frontend
 - Backend
 - 3 pages

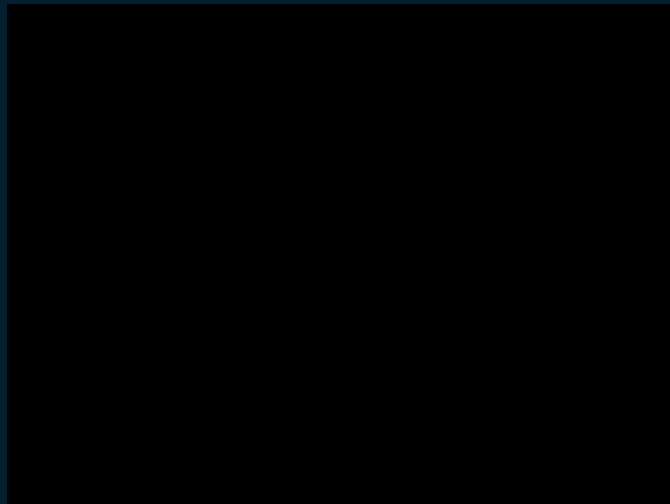


Thanks to Fatih: [Face Detection App Tutorial - GitHub](#)

6. Video



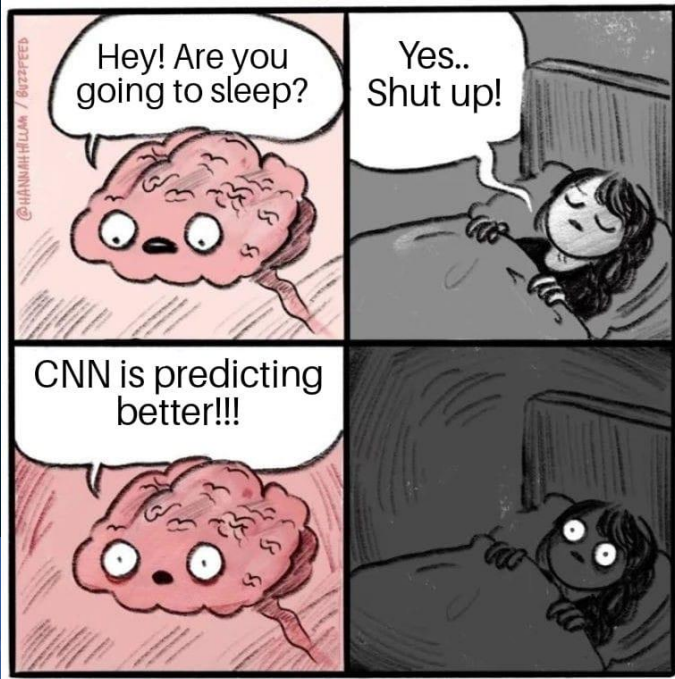
[Me - 26 years old](#)

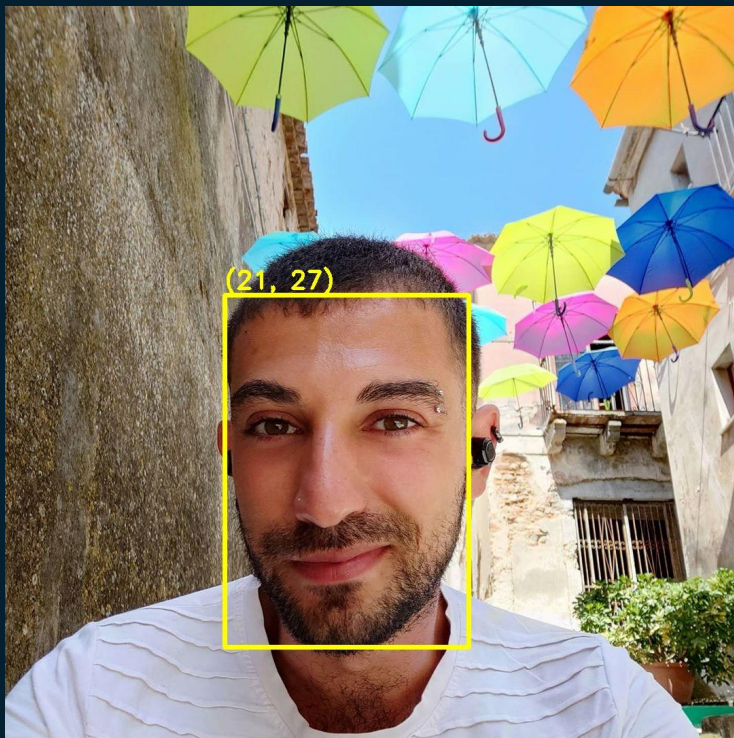


[My brother - 34 years old](#)

7. Conclusion

Convolutional Neural
Network looking at
Random Forest
Classifier's predictions





Thank
you
for
the
attention