Maskenerkennung

Lars Kolk
Jonah Blank

July 16, 2020

TU Dortmund

ML-Seminar

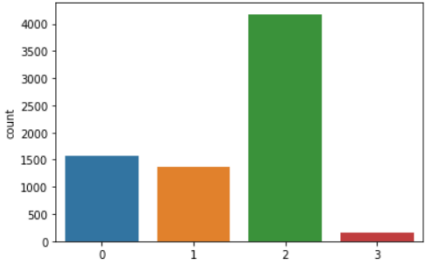
Fragestellung

- Grundlegende Fragestellung:
- “Kann ein neuronales Netz erkennen, ob eine Person eine Maske trägt?”

- Inhalt
  - Datensatz
  - Datenaufbereitung
  - Fully Connected Network vs Convolutional Network
  - Aussicht

Datensatz

- Quelle: Kaggle
- Lizenz: Public Domain (CC0)
- 6024 Bilder $\rightarrow 7271$ Gesichter
- 4 Oberklassen:
  - face_no_mask: 0
  - face_other_covering: 1
  - face_with_mask: 2
  - face_with_mask_incorrect: 3



Datenaufbereitung

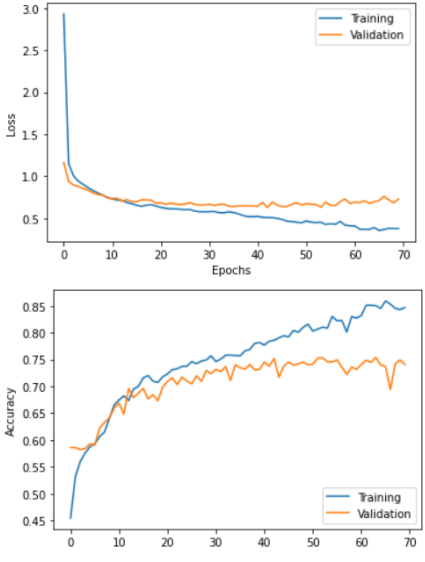
- Schneide Bilder auf Gesichter zu
- Bringe alle Bilder auf die gleiche Größe
  - 50×50 Pixel
- Berechne Matrizen der Bilder
- Wende MinMaxScalar an

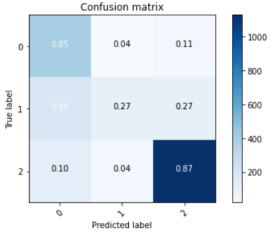
	A	B	C	D	E	F
1	name	x1	x2	y1	y2	classname
2	2756.png	69	126	294	392	face_with_mask
3	2756.png	505	10	723	283	face_with_mask
4	2756.png	75	252	264	390	mask_colorful
5	2756.png	521	136	711	277	mask_colorful
6	6098.jpg	360	85	728	653	face_no_mask
7	6427.png	218	98	577	580	face_with_mask_incr
8	6427.png	278	386	582	582	mask_surgical
9	4591.png	239	9	522	342	face_with_mask
10	4591.png	255	159	491	341	mask_colorful
11	5392.jpg	261	0	444	257	face_other_covering
12	5392.jpg	261	2	444	257	scarf_bandana
13	5525.jpg	262	18	484	319	face_no_mask
14	5525.jpg	49	58	191	247	face_no_mask
15	3911.png	49	26	389	476	face_no_mask
16	4287.png	160	312	588	595	mask_colorful
17	4287.png	166	201	592	336	eyeglasses
18	4287.png	80	0	629	599	face_with_mask
19	4893.png	179	8	664	599	face_with_mask
20	4893.png	185	214	584	596	mask_colorful
21	1987.jpg	365	67	464	199	face_with_mask
22	1987.jpg	72	150	103	191	face_no_mask

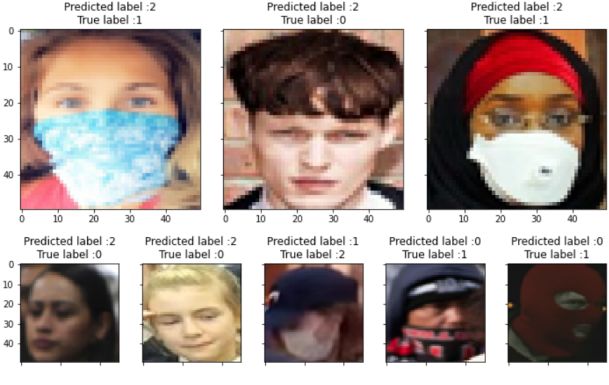
Fully Connected Network

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 7500)	0
dense (Dense)	(None, 1024)	7681024
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dense_3 (Dense)	(None, 64)	16448
dense_4 (Dense)	(None, 3)	195

dropout = 0.2




Confusion matrix
0.85
0.04
0.11
0.46
0.27
0.27
0.10
0.04
0.87
True label
Predicted label
0
1
2
1000
800
600
400
200


Predicted label :2
True label :1
Predicted label :2
True label :0
Predicted label :2
True label :1
Predicted label :2
True label :0
Predicted label :2
True label :0
Predicted label :1
True label :2
Predicted label :0
True label :1
Predicted label :0
True label :1

Convolutional Neural Network

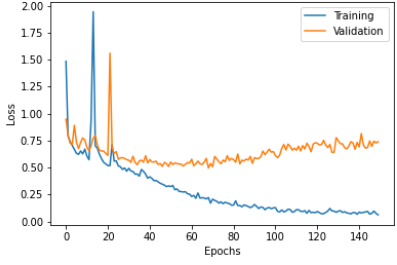
```
param_1 = {
    'nb1': [1, 2, 4],
    'nb2': [1, 2, 4],
    'filter': [32, 64, 128]
}

param_2 = {
    'kernelsize': [(2,2), (3,3), (6,6),(11,11)],
    'poolsize': [(2,2), (3,3), (4,4)],
}


param_3 = {
    'dropout': [0.3, 0.5, 0.7],
    'opti': ['RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']

}

param_4 = {
    'bs': [100, 150, 180, 210, 260],
    'lr': [1e-2, 1e-3, 1e-4, 1e-5]
}
```

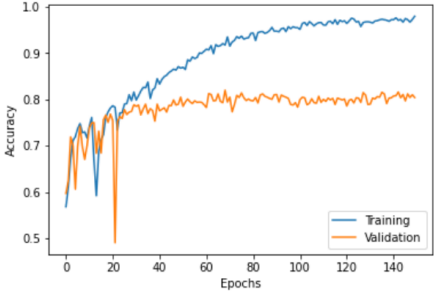
Convolutional Neural Network

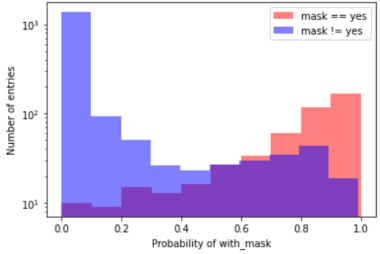
```
param_1 = {
    'nb1': [1, 2, 4],
    'nb2': [1, 2, 4],
    'filter': [32, 64, 128]
}

param_2 = {
    'kernelsize': [(2,2), (3,3), (6,6),(11,11)],
    'poolsize': [(2,2), (3,3), (4,4)],
}


param_3 = {
    'dropout': [0.3, 0.5, 0.7],
    'opti': ['RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']

}

param_4 = {
    'bs': [100, 150, 180, 210, 260],
    'lr': [1e-2, 1e-3, 1e-4, 1e-5]
}
```

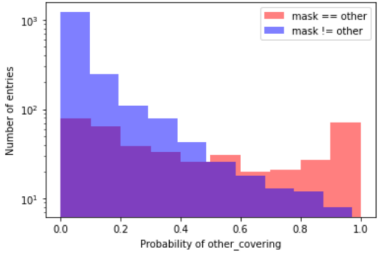
Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 48, 48, 512)	14336
dropout_20 (Dropout)	(None, 48, 48, 512)	0
max_pooling2d_4 (MaxPooling2	(None, 24, 24, 512)	0
dropout_21 (Dropout)	(None, 24, 24, 512)	0
conv2d_13 (Conv2D)	(None, 20, 20, 256)	3277056
dropout_22 (Dropout)	(None, 20, 20, 256)	0
conv2d_14 (Conv2D)	(None, 15, 15, 128)	1179776
dropout_23 (Dropout)	(None, 15, 15, 128)	0
flatten_4 (Flatten)	(None, 28800)	0
dropout_24 (Dropout)	(None, 28800)	0
dense_4 (Dense)	(None, 3)	86403

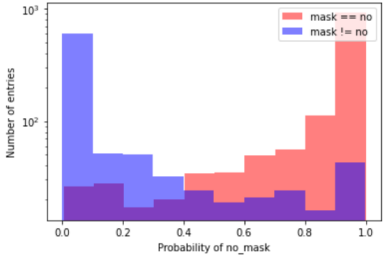
Total params: 4,557,571
Trainable params: 4,557,571
Non-trainable params: 0

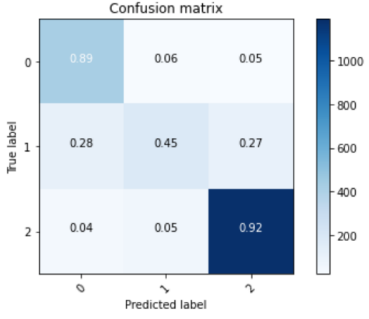

Training
Validation
Loss
Epochs
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
100
120
140


Training
Validation
Accuracy
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0
20
40
60
80
100
120
140


mask == yes
mask != yes
Number of entries
Probability of with_mask


mask == other
mask != other
Number of entries
Probability of other_covering


mask == no
mask != no
Number of entries
Probability of no_mask


Confusion matrix
0.89
0.06
0.05
0.28
0.45
0.27
0.04
0.05
0.92
True label
Predicted label