Emotionserkennung mittels CNN in tensorflow

Erstellen eines CNN

Test 0:
Modell:

```
  Layer (type)                 Output Shape              Param #
=================================================================
    conv2d (Conv2D)              (None, 46, 46, 64)        640

    conv2d_1 (Conv2D)            (None, 44, 44, 128)       73856

    conv2d_2 (Conv2D)            (None, 42, 42, 32)        36896

    conv2d_3 (Conv2D)            (None, 40, 40, 16)        4624

    conv2d_4 (Conv2D)            (None, 38, 38, 8)         1160

    flatten (Flatten)            (None, 11552)             0

    dense (Dense)                (None, 64)                739392

    dense_1 (Dense)              (None, 7)                 455


=================================================================
    Total params: 857,023
    Trainable params: 857,023
    Non-trainable params: 0
```

learning rate: 0.01
batch size: 32
epochs: 200

Das Training stoppte bereits nach der ersten Epoche mit einer accuracy von 25%, wahrscheinlich, da sich das Modell auf den besten Mittelwert trainiert hatte.
Daher reduzierte ich die learning rate und fügte Pooling layers und Dropout layers hinzu.

Test 1:
Modell:

```
  Layer (type)                 Output Shape              Param #
=================================================================
    conv2d (Conv2D)              (None, 46, 46, 64)        640

    dropout (Dropout)            (None, 46, 46, 64)        0

    max_pooling2d (MaxPooling2D  (None, 15, 15, 64)        0
    )

    conv2d_1 (Conv2D)            (None, 13, 13, 128)       73856

    conv2d_2 (Conv2D)            (None, 11, 11, 32)        36896

    dropout_1 (Dropout)          (None, 11, 11, 32)        0

    max_pooling2d_1 (MaxPooling  (None, 5, 5, 32)          0
    2D)

    conv2d_3 (Conv2D)            (None, 3, 3, 16)          4624

    conv2d_4 (Conv2D)            (None, 1, 1, 8)           1160

    flatten (Flatten)            (None, 8)                 0

    dense (Dense)                (None, 64)                576

    dropout_2 (Dropout)          (None, 64)                0

    dense_1 (Dense)              (None, 7)                 455


=================================================================
    Total params: 118,207
    Trainable params: 118,207
    Non-trainable params: 0
```

learning rate: 0.001
batch size: 32
epochs: 200

Das Modell hörte nach 16 Epochen auf, sich zu verbessern und erreichte eine accuracy von 25.11%
Ich erhöhte die learning rate und verdoppelte die Anzahl der Layer im Modell.

Test 2:
Modell:

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 64)	640
dropout (Dropout)	(None, 46, 46, 64)	0
max_pooling2d (MaxPooling2D)	(None, 15, 15, 64)	0
conv2d_1 (Conv2D)	(None, 13, 13, 128)	73856
conv2d_2 (Conv2D)	(None, 11, 11, 32)	36896
dropout_1 (Dropout)	(None, 11, 11, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 32)	0
conv2d_3 (Conv2D)	(None, 3, 3, 16)	4624
conv2d_4 (Conv2D)	(None, 1, 1, 8)	1160
up_sampling2d (UpSampling2D)	(None, 46, 46, 8)	0
conv2d_5 (Conv2D)	(None, 44, 44, 64)	4672
dropout_2 (Dropout)	(None, 44, 44, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_6 (Conv2D)	(None, 12, 12, 128)	73856
conv2d_7 (Conv2D)	(None, 10, 10, 32)	36896
dropout_3 (Dropout)	(None, 10, 10, 32)	0
max_pooling2d_3 (MaxPooling2D)	(None, 5, 5, 32)	0
conv2d_8 (Conv2D)	(None, 3, 3, 16)	4624
conv2d_9 (Conv2D)	(None, 1, 1, 8)	1160
flatten (Flatten)	(None, 8)	0
dense (Dense)	(None, 256)	2304
dense_1 (Dense)	(None, 64)	16448
dropout_4 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 7)	455

```
=================================================================
    Total params: 257,591
    Trainable params: 257,591
    Non-trainable params: 0
```

learning rate: 0.005
batch size: 32
epochs: 200

Das Modell erreichte nach 8 Epochen eine accuracy von 25.1% und verbesserte sich danach nicht weiter.
Ich baute das Modell leicht um und fügte BatchNormalization layers hinzu

Ich überspringe hier mehrere nicht funktionale Modelle, die alle eine accuracy von 25% nicht maßgeblich überschritten haben. Die learing rate (0.001), batch size (32) und Epochenanzahl (200) waren für alle versuche gleich und ich habe mittels versuch und Irrtum dass Modell umgeordnet.

Im folgenden ist das erste funktionierende Modell:

Test 24:
Model:

```
  Layer (type)                 Output Shape              Param #
=================================================================
    conv2d (Conv2D)              (None, 46, 46, 64)        640

    batch_normalization (BatchN  (None, 46, 46, 64)        256
    ormalization)

    max_pooling2d (MaxPooling2D  (None, 23, 23, 64)        0
    )

    dropout (Dropout)            (None, 23, 23, 64)        0

    conv2d_1 (Conv2D)            (None, 21, 21, 128)       73856

    max_pooling2d_1 (MaxPooling  (None, 10, 10, 128)       0
    2D)

    conv2d_2 (Conv2D)            (None, 8, 8, 32)          36896

    batch_normalization_1 (Batc  (None, 8, 8, 32)          128
    hNormalization)

    max_pooling2d_2 (MaxPooling  (None, 4, 4, 32)          0
    2D)

    dropout_1 (Dropout)          (None, 4, 4, 32)          0

    up_sampling2d (UpSampling2D  (None, 16, 16, 32)        0
    )

    conv2d_3 (Conv2D)            (None, 14, 14, 16)        4624

    max_pooling2d_3 (MaxPooling  (None, 7, 7, 16)          0
    2D)

    conv2d_4 (Conv2D)            (None, 5, 5, 8)           1160

    batch_normalization_2 (Batc  (None, 5, 5, 8)           32
    hNormalization)

    max_pooling2d_4 (MaxPooling  (None, 2, 2, 8)           0
    2D)

    up_sampling2d_1 (UpSampling  (None, 16, 16, 8)         0
    2D)

    dropout_2 (Dropout)          (None, 16, 16, 8)         0

    conv2d_5 (Conv2D)            (None, 14, 14, 128)       9344

    max_pooling2d_5 (MaxPooling  (None, 7, 7, 128)         0
    2D)
```

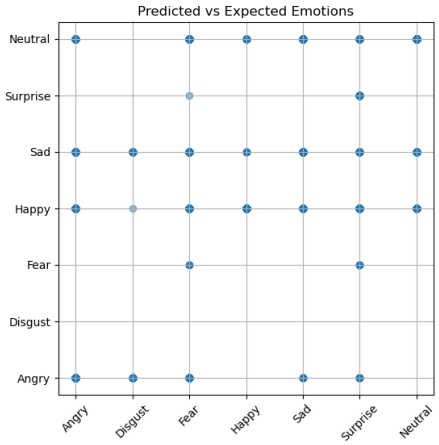
conv2d_6 (Conv2D)	(None, 5, 5, 236)	272108
up_sampling2d_2 (UpSampling 2D)	(None, 20, 20, 236)	0
conv2d_7 (Conv2D)	(None, 18, 18, 16)	34000
max_pooling2d_6 (MaxPooling 2D)	(None, 9, 9, 16)	0
conv2d_8 (Conv2D)	(None, 7, 7, 8)	1160
batch_normalization_3 (Batc hNormalization)	(None, 7, 7, 8)	32
max_pooling2d_7 (MaxPooling 2D)	(None, 3, 3, 8)	0
up_sampling2d_3 (UpSampling 2D)	(None, 24, 24, 8)	0
dropout_3 (Dropout)	(None, 24, 24, 8)	0
conv2d_9 (Conv2D)	(None, 22, 22, 4)	292
max_pooling2d_8 (MaxPooling 2D)	(None, 11, 11, 4)	0
conv2d_10 (Conv2D)	(None, 9, 9, 2)	74
up_sampling2d_4 (UpSampling 2D)	(None, 36, 36, 2)	0
batch_normalization_4 (Batc hNormalization)	(None, 36, 36, 2)	8
max_pooling2d_9 (MaxPooling 2D)	(None, 18, 18, 2)	0
dropout_4 (Dropout)	(None, 18, 18, 2)	0
conv2d_11 (Conv2D)	(None, 16, 16, 4)	76
max_pooling2d_10 (MaxPoolin g2D)	(None, 8, 8, 4)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 256)	65792
dense_1 (Dense)	(None, 64)	16448
batch_normalization_5 (Batc hNormalization)	(None, 64)	256
dropout_5 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 7)	455

```
=================================================================
    Total params: 517,637
    Trainable params: 517,281
    Non-trainable params: 356
```

learning rate: 0.001
batch size: 32
epochs: 200

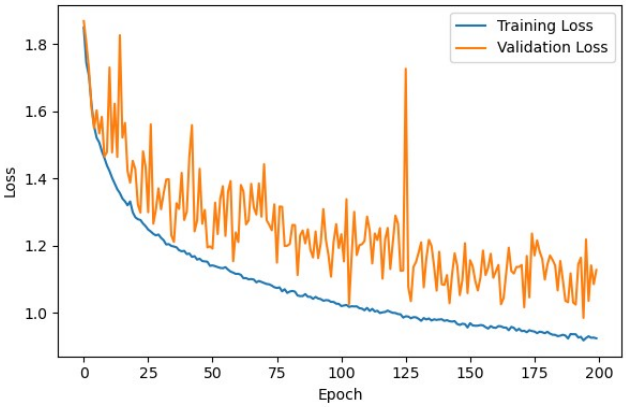
Dieses Modell erreichte eine accuracy von 56% und hatte am ende der 200 Epochen das Training noch nicht abgeschlossen.
Ich erhöhte daher die learning rate.

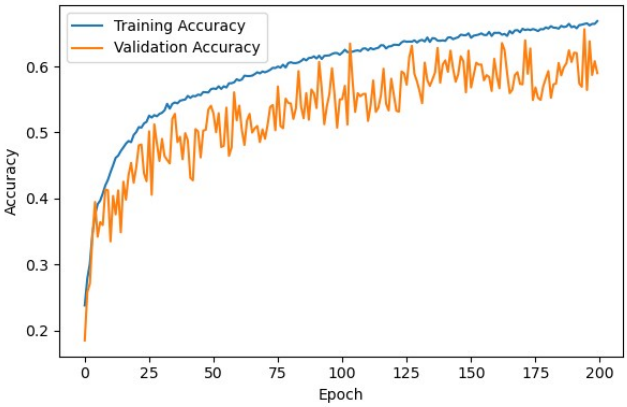
Test 25:
Modell: wie 24



learning rate: 0.005
batch size: 32
epochs: 200

Das Modell erreichte am ende des Trainings eine test-accuracy von 59.11%
Allerdings erreichte die validation-accuracy schon bei Epoche 150 ein Plateau und oszillierte zwischen 56% und 64%.
Ich verringerte daher die Epochenanzahl und erhöhte die learning rate weiter.

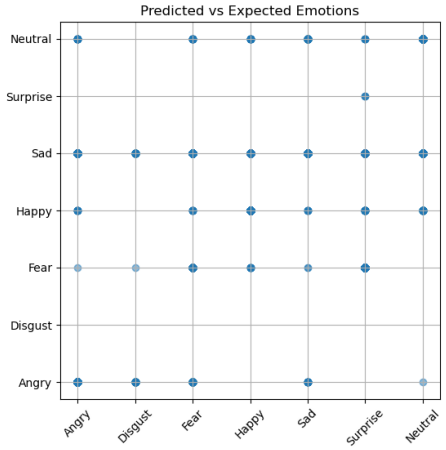




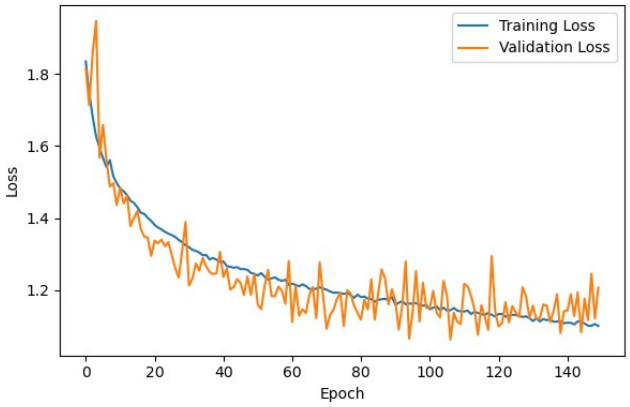
Test 26:
Modell: wie 24
learning rate: 0.01
batch size: 32
epochs: 200

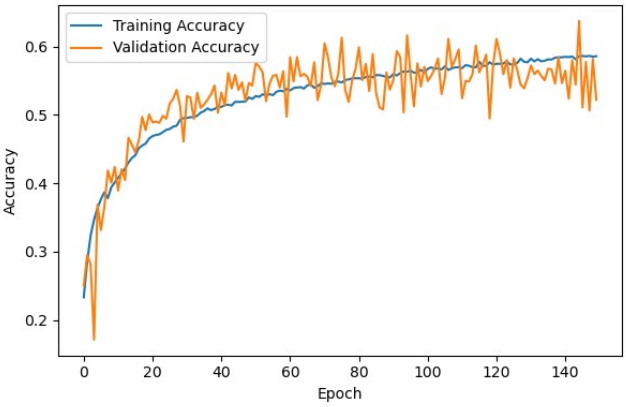
Das Modell oszilliert zwischen einer validation accuracy von 15% bis 30%
Ich senkte daher die learning rate wieder und verringerte die batch size, um die Oszillation zu verringern

Test 27:
Modell: wie 24
learning rate: 0.005
batch size: 16
epochs: 150

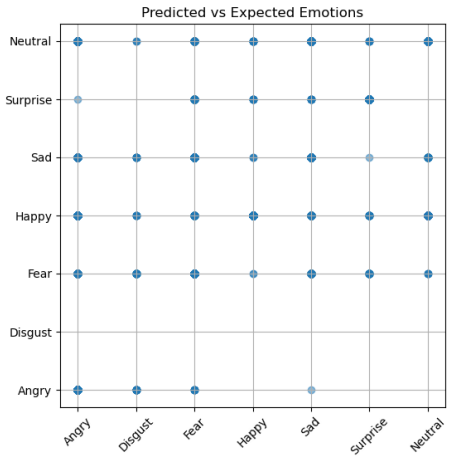


Das Modell erreichte nur noch eine test-accuracy von 53.13% und zeigt ähnliche Oszillation wie 25.
Ich erhöhte also die batch size wider, in der Hoffnung, dadurch eine höhere accuracy zu erreichen.

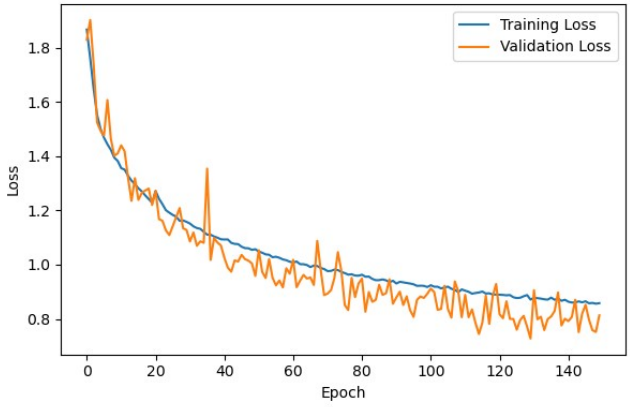


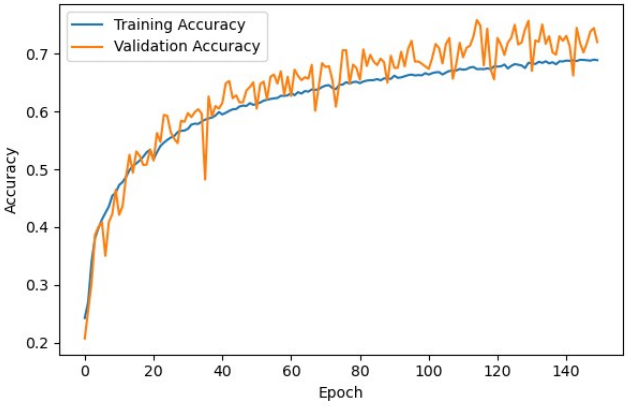


Test 28:
Modell: wie 24
learning rate: 0.005
batch size: 64
epochs: 150



Das Modell erreicht eine test-accuracy von 71.71%. Bereits nach 50 Epochen wird eine validation-accuracy von über 60% erreicht.
Daher reduziere ich die Epochenanzahl auf 80. Dadurch geht zwar Genauigkeit verloren, jedoch wird der Zeitaufwand für das Training reduziert.





Test 29:
Modell: wie 24
learning rate: 0.005
batch size: 64
epochs: 80

Das Modell erreicht eine test-accuracy von 67.74%.
Das scatter-diagramm offenbart allerdings noch eine hohe Ungenauigkeit der Daten.
Da das Modell die Aufgabenstellung erfüllt und vergleichsweise schnell trainiert werden kann, werde ich mit diesem Modell weiter arbeiten.

Test mit gespiegelten Daten

Test 30:
Modell und Parameter: wie 29

Mit gespiegelten Daten wird eine test-accuracy von 63.75% erreicht. Das ist eine geringfügig geringere Genauigkeit als ohne gespiegelte Daten, diese leichte Abweichung ist aber höchstwahrscheinlich zu einem signifikanten Anteil auf zufällige Abweichungen zurück zu führen. Der scatter-plot wiederum wirkt kohärenter als bei Test 29. Der Loss-plot suggeriert, dass das Training noch für mehrere Epochen fortgeführt werden könnte, um die Genauigkeit zu erhöhen.