Movie Genre by Poster

Transfer Learning mit ResNet50V2

Fragestellung

Kann man Filmgenres anhand des Posters vorhersagen?

Abbildung 1: Filmposter

poster-genre-prediction | main

main

poster_predictor.ipynb | pyplot.py | categorical.py | .gitignore

Code | Managed Jupyter server: auto-start | Python 3 | Trusted

3 26 21

Genre Predictor by Movie Poster

This project aims to predict the genres of a movie just by analyzing it's poster image.

Author: Alexander Ehrenhöfer | s0580781
Dataset src: https://www.kaggle.com/datasets/raman77768/movie-classifier

In 1

```
import tensorflow as tf
import cv2
import pandas as pd
import copy
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from keras import utils
from sklearn.model_selection import train_test_split
from keras.preprocessing.image import ImageDataGenerator
from keras.applications import ResNet50V2
from keras.applications import ConvNeXtSmall
from keras.applications import EfficientNetB0
from keras.applications import MobileNetV2
from keras.models import Sequential
from keras import backend as K
from keras.layers import *
from keras.optimizers import Adam
from keras.models import load_model
from keras.callbacks import ModelCheckpoint, EarlyStopping
from sklearn.metrics import roc_curve
```

Executed at 2024.01.15 09:17:26 in 8s 361ms

```
WARNING:tensorflow:From C:\Users\alex\anaconda3\envs\new-poster-genre-prediction\lib\site-packages\keras\src\losses.py:2976: The name
tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

Data Preparation

poster-genre-prediction > poster_predictor.ipynb

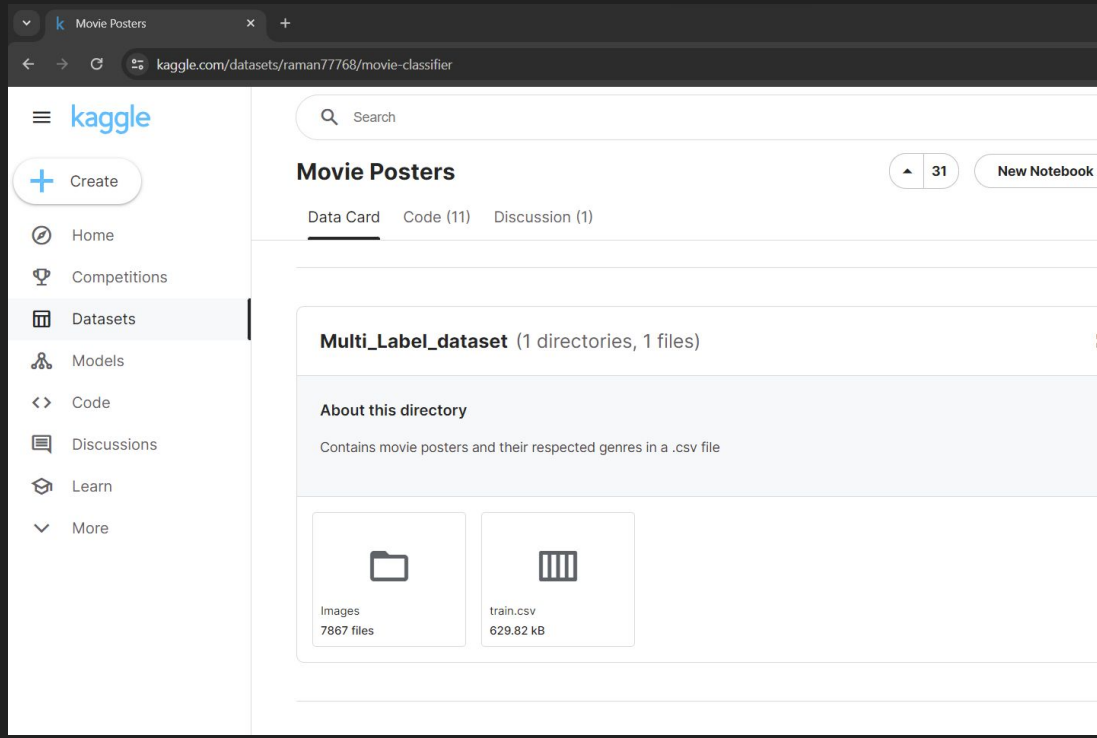
CRLF UTF-8 4 spaces new-poster-genre-prediction

Inhalt

- Datenanalyse
- Transferlearning + Fine Tuning
- Vorgehensweise / Modellauswahl
- Ergebnis
- CNN Filter Visualisierung

Daten

- Download von Kaggle
- Basierend auf IMDB
- 7867 Poster
- 25 Genre



Datenanalyse

```
print('---- basic structure of the data ----')
df.head(5)
```

Executed at 2023.12.30 10:53:30 in 75ms

```
---- basic structure of the data ----
```

5 rows | 5 rows × 30 columns pd.Dataframe

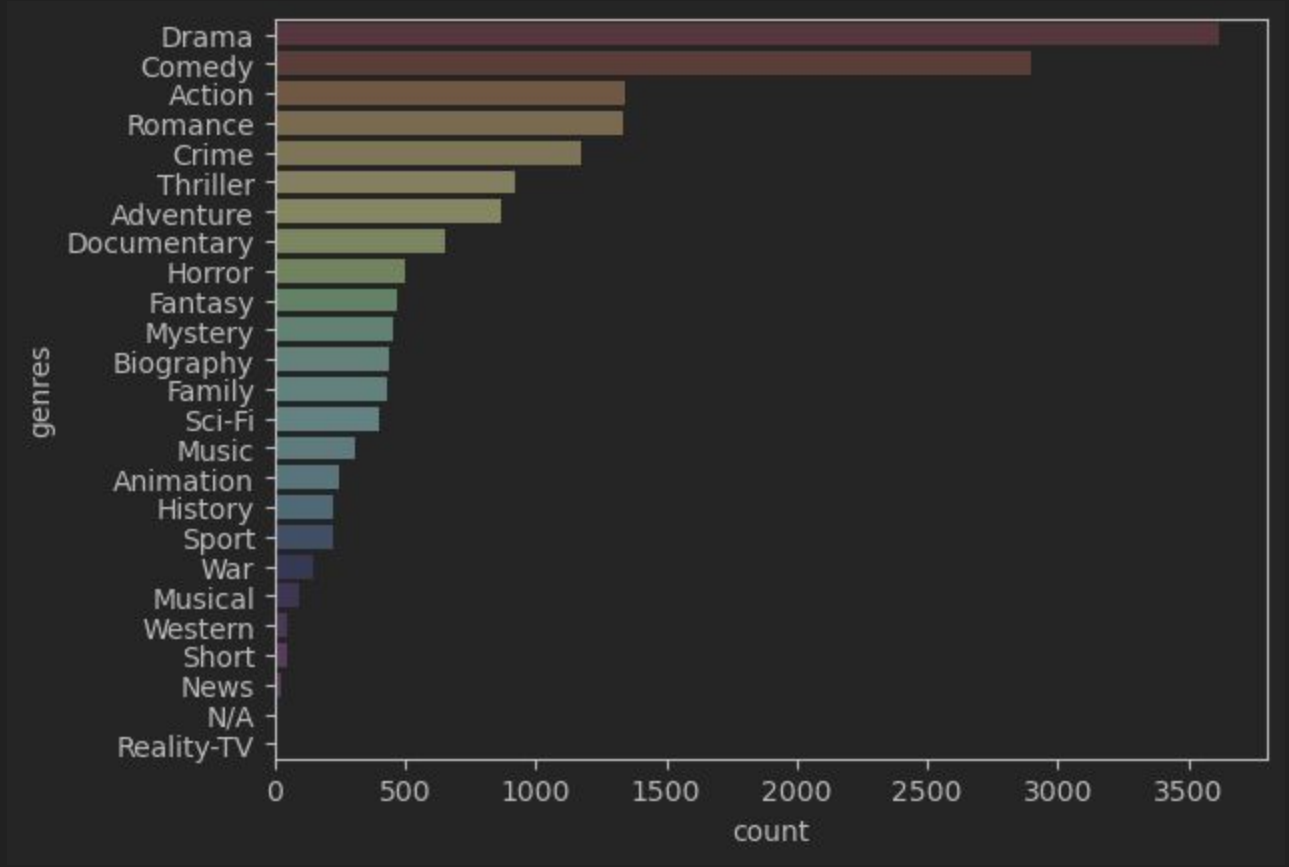
	Id	Genre	Action	Adventure
0	tt0086425	['Comedy', 'Drama']	0	0
1	tt0085549	['Drama', 'Romance', 'Music']	0	0
2	tt0086465	['Comedy']	0	0
3	tt0086567	['Sci-Fi', 'Thriller']	0	0
4	tt0086034	['Action', 'Adventure', 'Thriller']	1	1

```
df.dtypes
```

Executed at 2023.12.30 10:53:30 in 289ms

1-10 | Length: 30, dtype: object pd.Series

	<unnamed>
Id	object
Genre	object
Action	int64
Adventure	int64
Animation	int64
Biography	int64
Comedy	int64
Crime	int64
Documentary	int64


Drama
Comedy
Action
Romance
Crime
Thriller
Adventure
Documentary
Horror
Fantasy
Mystery
Biography
Family
Sci-Fi
Music
Animation
History
Sport
War
Musical
Western
Short
News
N/A
Reality-TV
genres
0
500
1000
1500
2000
2500
3000
3500
count

Transfer Learning

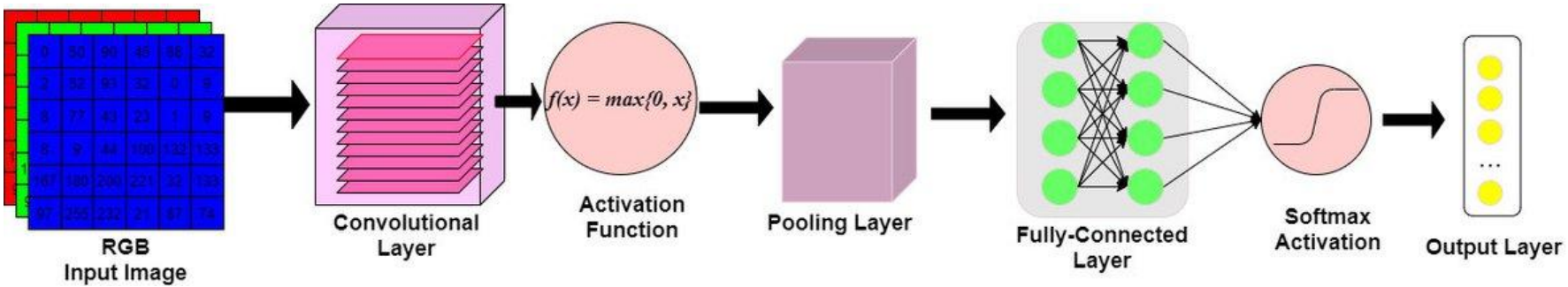


Abbildung 2: CNN

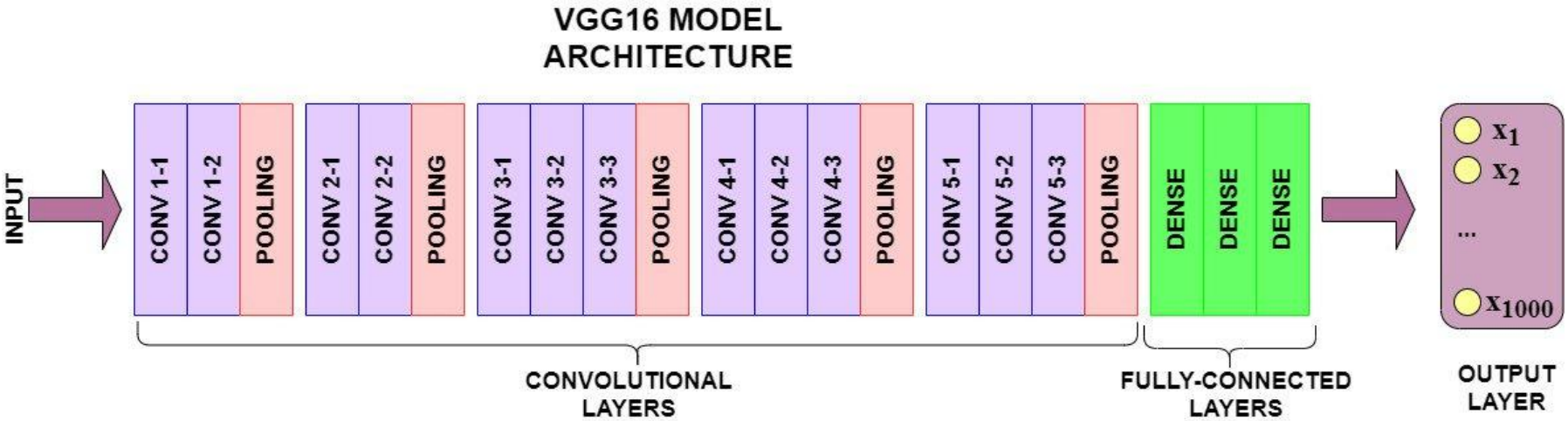


Abbildung 3: CNN Layer

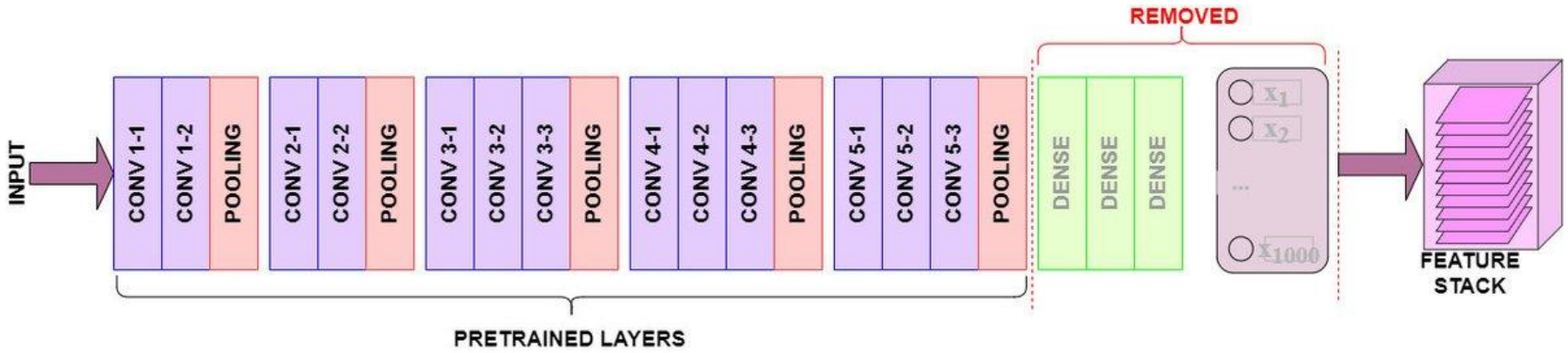


Abbildung 4: CNN Top Layer entfernen

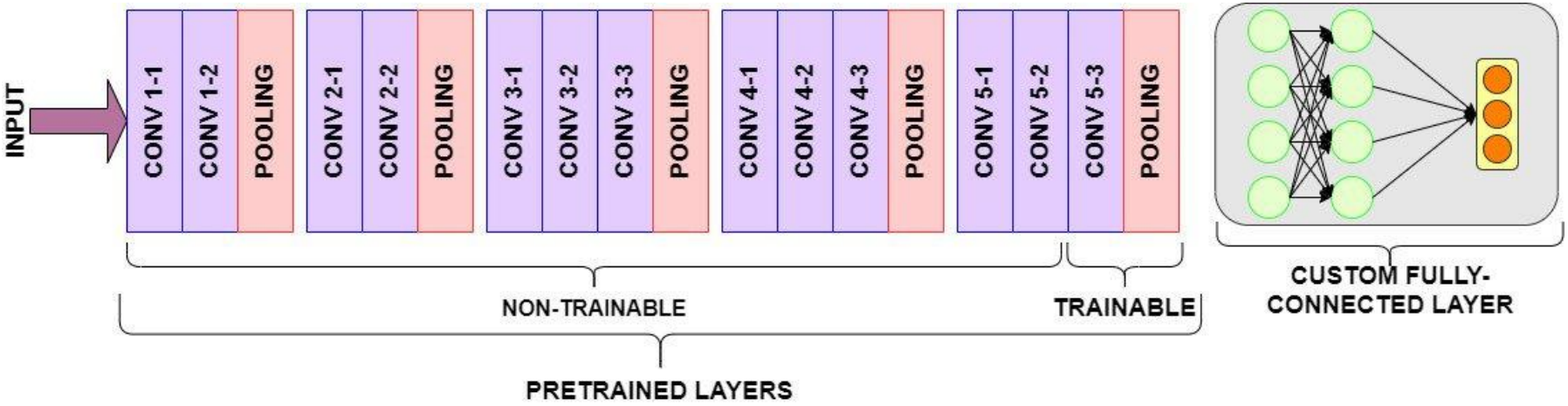


Abbildung 5: CNN Fine Tuning

Vorgehensweise

Direkte Vorhersage mit Originaldaten

- Transfer Learning mit vorrainiertem VGG16
- Input: Original Bild (Pixel) Daten
- Output: 25 Neuronen (pro Genre)
- Problem: Extremes Overfitting

```
53
54  #Shuffle Array
55  df = df.sample(frac=1)
56
57  # Input NP Array (Image Data directly as pixel values)
58  width = 224
59  height = 224
60  x_data = []
61  for index, row in df.iterrows():
62      img = image.load_img(row['image_path'],target_size=(width,height,3))
63      img = image.img_to_array(img)
64      img = img/255.0
65      x_data.append(img)
66
67  x_data = np.array(x_data)
68
69  # Output Array --> Genre Matrix
70  y_data = df[genres_to_predict]
71
72  x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.2)
73  x_train, x_val, y_train,y_val = train_test_split(x_train, y_train, test_size=0.2)
```

Executed at 2023.12.21 21:28:56 in 6s 42ms

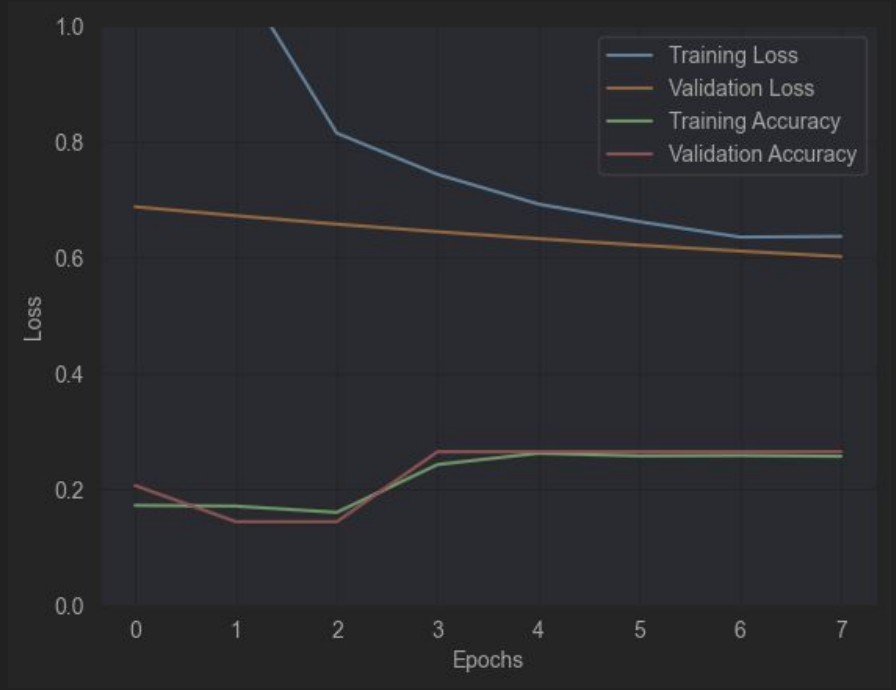

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Loss
0
5
10
15
20
25
30
Epochs
Training Loss
Validation Loss
Training Accuracy
Validation Accuracy

Direkte Vorhersage mit Data Augmentation

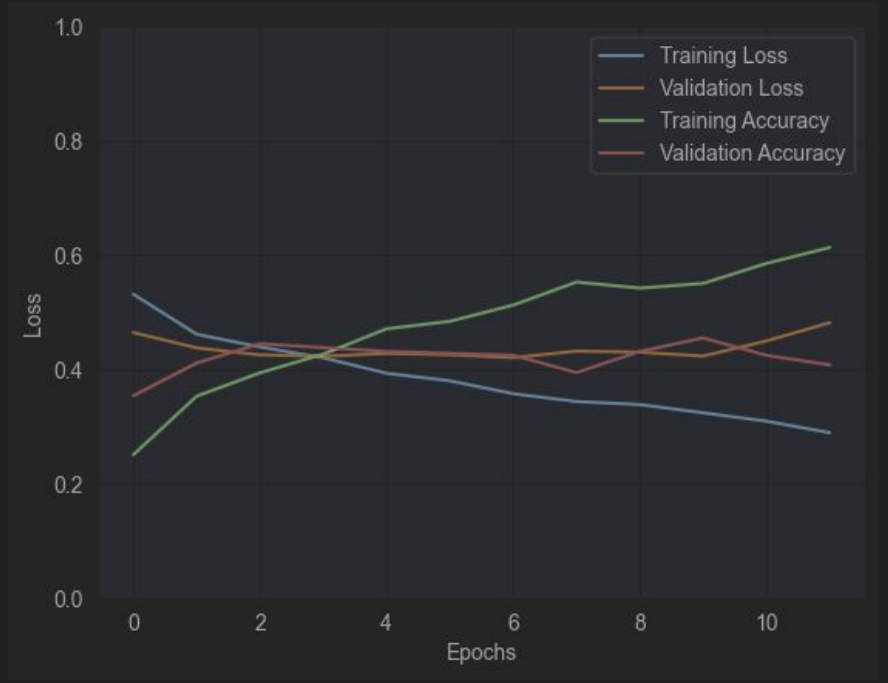
- Gleiches Modell wie zuvor
- Data Augmentation mit Keras ImageDataGenerator
- Problem: Accuracy sehr schlecht

```
# Create a data generator with data augmentation for training
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    validation_split=0.15,
    horizontal_flip=False
)
```

Transfer Learning



Eigenes Model



Neuer Ansatz: Nur pro Modell 1 Genre vorhersagen

- Saubere Data Balance herstellen
- Ja/Nein Vorhersage

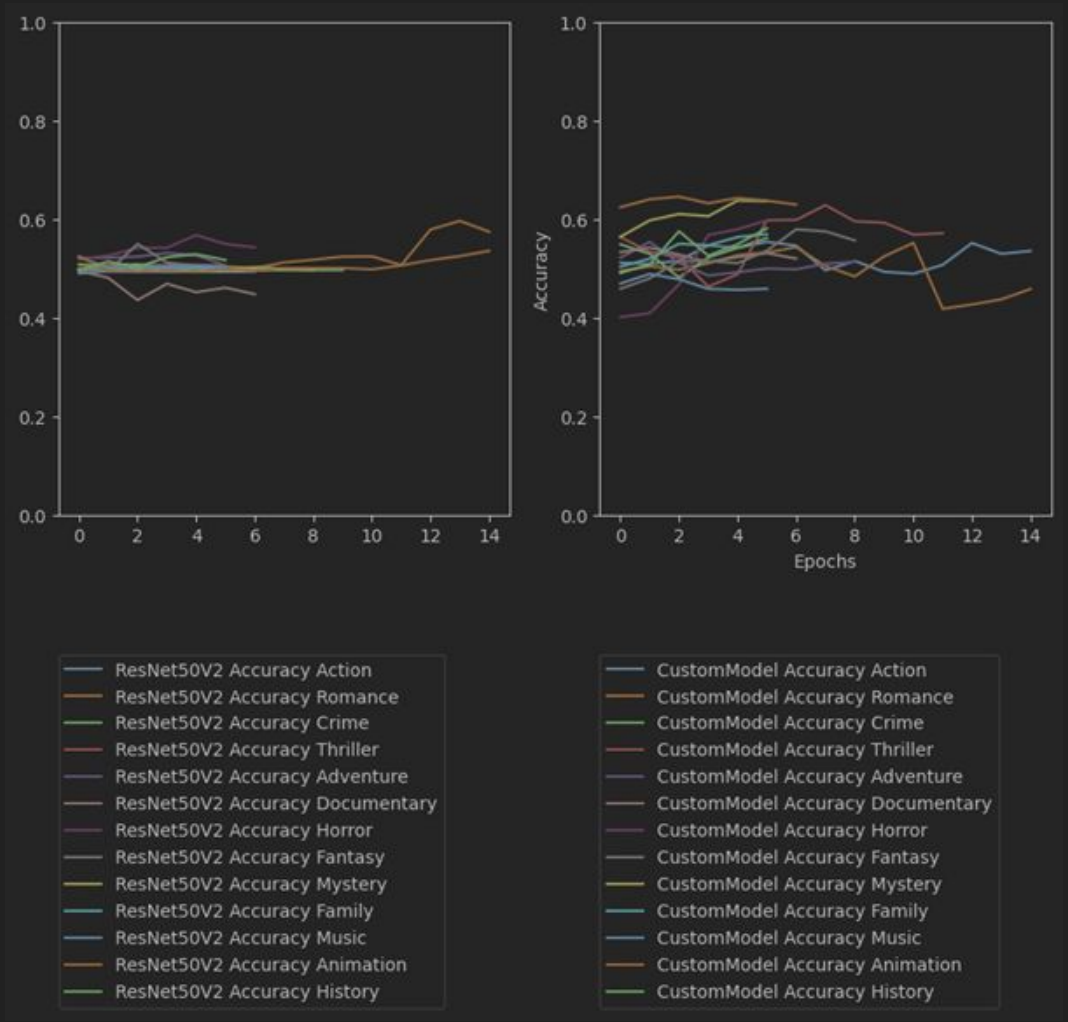
```
def get_raw_data_by_genre(data_frame, genre):
    shuffled_frame = copy.copy(data_frame.sample(frac=1))
    positive_data = shuffled_frame[shuffled_frame[genre] == 1]
    negative_data = shuffled_frame[shuffled_frame[genre] == 0]

    # make sure data is balanced
    if len(positive_data) > len(negative_data):
        positive_data = positive_data[0:len(negative_data)]
    else:
        negative_data = negative_data[0:len(positive_data)]

    complete_data = pd.concat([positive_data, negative_data])
    complete_data = complete_data.sample(frac=1)

    print('--- data for genre specific model ---')
    print('num ' + genre + ' == 1: ' + str(len(complete_data[complete_data[genre] == 1])))
    print('num ' + genre + ' == 0: ' + str(len(complete_data[complete_data[genre] == 0])))
    print ('--- --- ---')
```

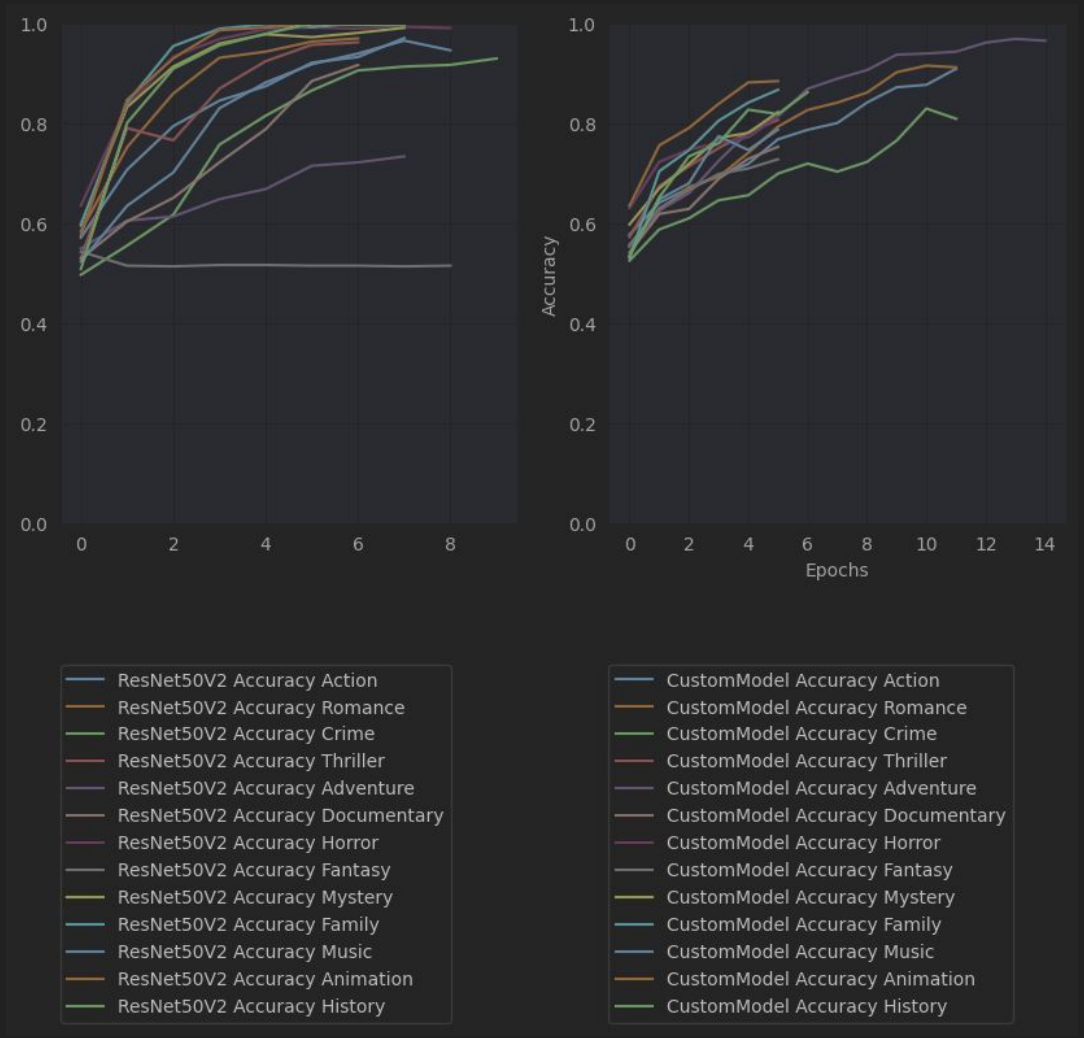
Resnet

```python
def get_resnet_model(fine_tune=2):
    #based on https://www.learndatasci.com/tutorials/hands-on-transfer-learning-keras/
    # Load the pre-trained Resnet model
    base_model = ResNet50V2(include_top=False, input_shape=(300, 300, 3))

    # Freeze the layers of the pre-trained model (exclude fine tuning layers)
    if fine_tune > 0:
        for layer in base_model.layers[:-fine_tune]:
                layer.trainable = False
    else:
        for layer in base_model.layers:
                layer.trainable = False

    # Create a new model for genre prediction
    return Sequential([
        base_model,
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.1),
        Dense(1, activation='sigmoid')
    ])

```

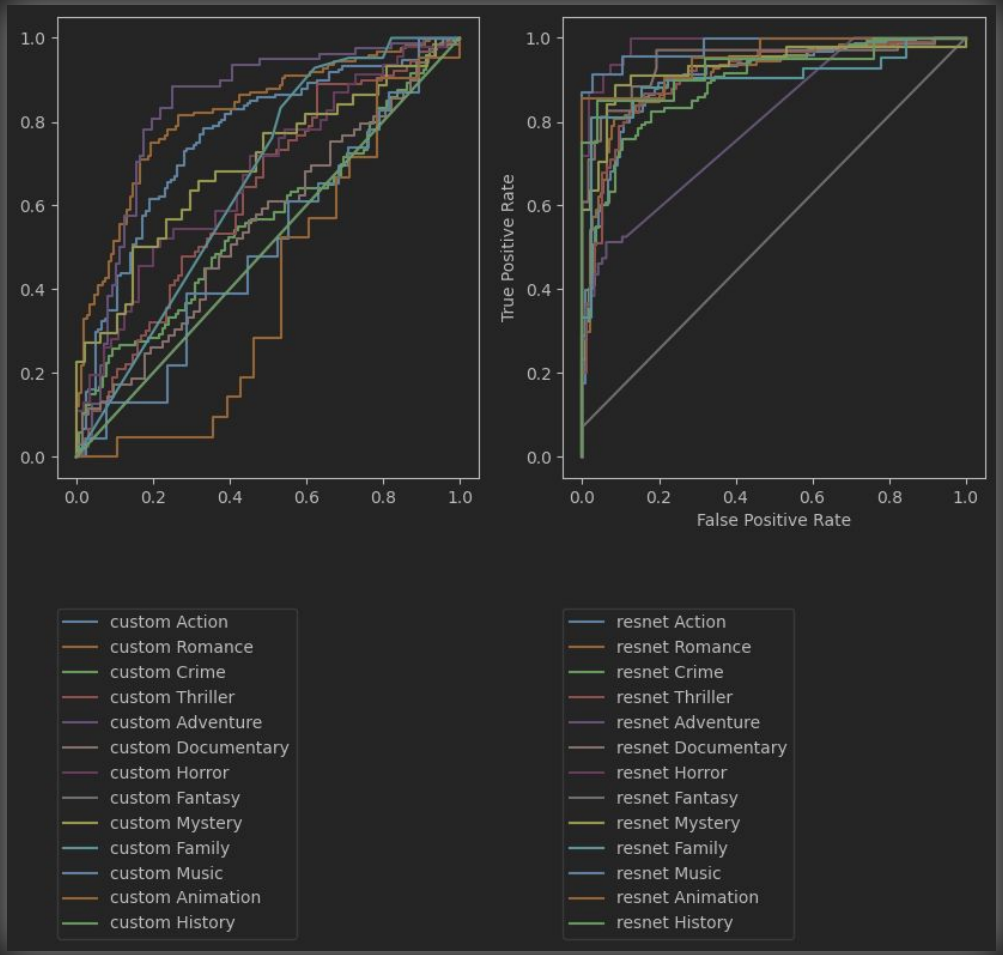
Custom Model

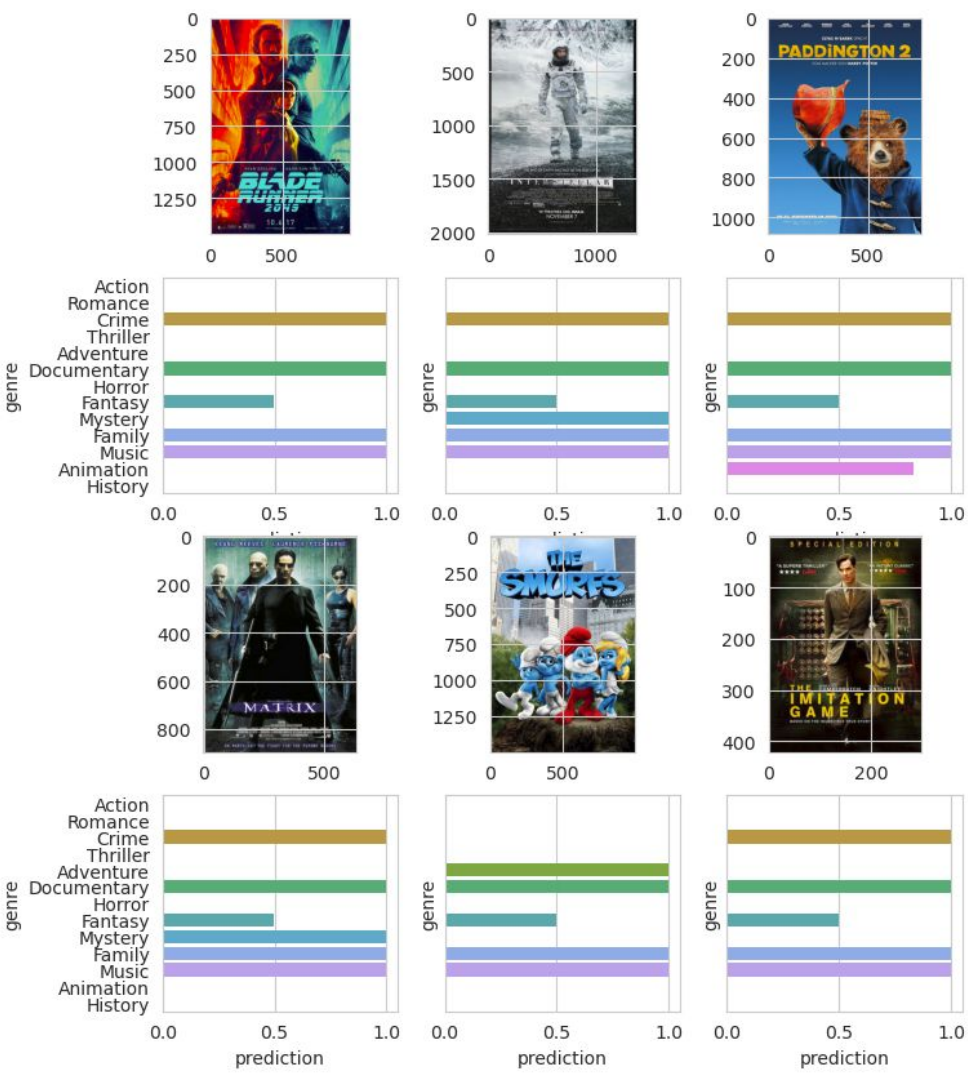
```python
def get_poster_model():
    model = Sequential()
    model.add(Conv2D(16,kernel_size=(3,3),activation='relu',input_shape=(300, 300, 3)))
    model.add(BatchNormalization())
    model.add(MaxPool2D(2,2))
    model.add(Dropout(0.3))

    model.add(Conv2D(32,kernel_size=(3,3),activation='relu'))
    model.add(BatchNormalization())
    model.add(MaxPool2D(2,2))
    model.add(Dropout(0.3))

    model.add(Conv2D(64,kernel_size=(3,3),activation='relu'))
    model.add(BatchNormalization())
    model.add(MaxPool2D(2,2))
    model.add(Dropout(0.4))

    model.add(Flatten())

    model.add(Dense(128,activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(1,activation='sigmoid'))
    return model
```


1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
Accuracy
Epochs
ResNet50V2 Accuracy Action
ResNet50V2 Accuracy Romance
ResNet50V2 Accuracy Crime
ResNet50V2 Accuracy Thriller
ResNet50V2 Accuracy Adventure
ResNet50V2 Accuracy Documentary
ResNet50V2 Accuracy Horror
ResNet50V2 Accuracy Fantasy
ResNet50V2 Accuracy Mystery
ResNet50V2 Accuracy Family
ResNet50V2 Accuracy Music
ResNet50V2 Accuracy Animation
ResNet50V2 Accuracy History
CustomModel Accuracy Action
CustomModel Accuracy Romance
CustomModel Accuracy Crime
CustomModel Accuracy Thriller
CustomModel Accuracy Adventure
CustomModel Accuracy Documentary
CustomModel Accuracy Horror
CustomModel Accuracy Fantasy
CustomModel Accuracy Mystery
CustomModel Accuracy Family
CustomModel Accuracy Music
CustomModel Accuracy Animation
CustomModel Accuracy History


1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
Epochs
ResNet50V2 Accuracy Action
ResNet50V2 Accuracy Romance
ResNet50V2 Accuracy Crime
ResNet50V2 Accuracy Thriller
ResNet50V2 Accuracy Adventure
ResNet50V2 Accuracy Documentary
ResNet50V2 Accuracy Horror
ResNet50V2 Accuracy Fantasy
ResNet50V2 Accuracy Mystery
ResNet50V2 Accuracy Family
ResNet50V2 Accuracy Music
ResNet50V2 Accuracy Animation
ResNet50V2 Accuracy History
CustomModel Accuracy Action
CustomModel Accuracy Romance
CustomModel Accuracy Crime
CustomModel Accuracy Thriller
CustomModel Accuracy Adventure
CustomModel Accuracy Documentary
CustomModel Accuracy Horror
CustomModel Accuracy Fantasy
CustomModel Accuracy Mystery
CustomModel Accuracy Family
CustomModel Accuracy Music
CustomModel Accuracy Animation
CustomModel Accuracy History

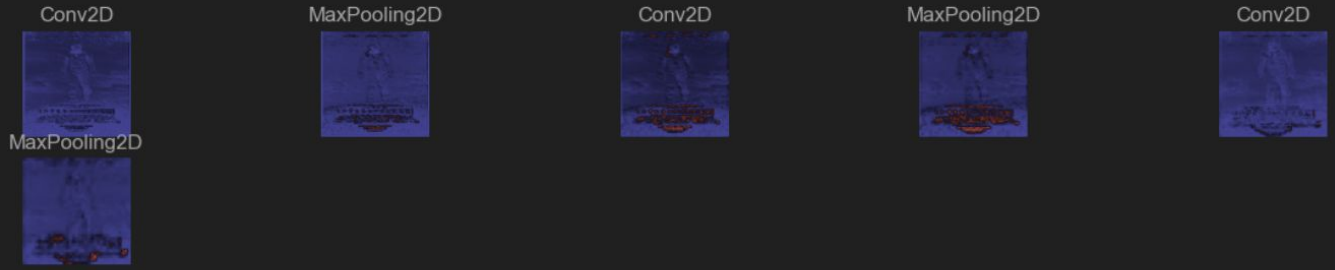
ROC Kurve


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
True Positive Rate
False Positive Rate
custom Action
custom Romance
custom Crime
custom Thriller
custom Adventure
custom Documentary
custom Horror
custom Fantasy
custom Mystery
custom Family
custom Music
custom Animation
custom History
resnet Action
resnet Romance
resnet Crime
resnet Thriller
resnet Adventure
resnet Documentary
resnet Horror
resnet Fantasy
resnet Mystery
resnet Family
resnet Music
resnet Animation
resnet History

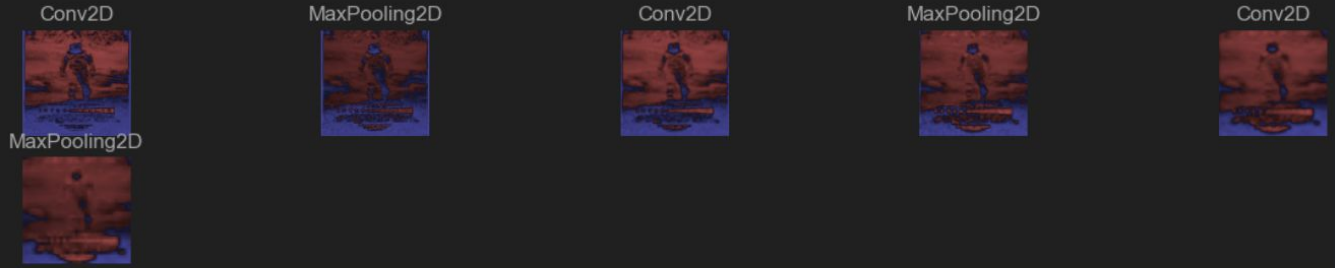

BLADE RUNNER 2049
INTERSTELLAR
PADDINGTON 2
MATRIX
THE SMURFS
IMITATION GAME
Action
Romance
Crime
Thriller
Adventure
Documentary
Horror
Fantasy
Mystery
Family
Music
Animation
History
genre
prediction

Visualisierung der Filter

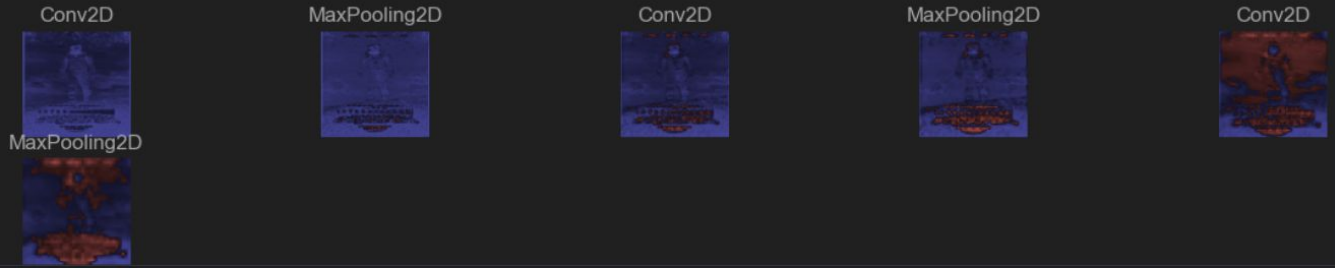
Model: Custom, Genre: Adventure



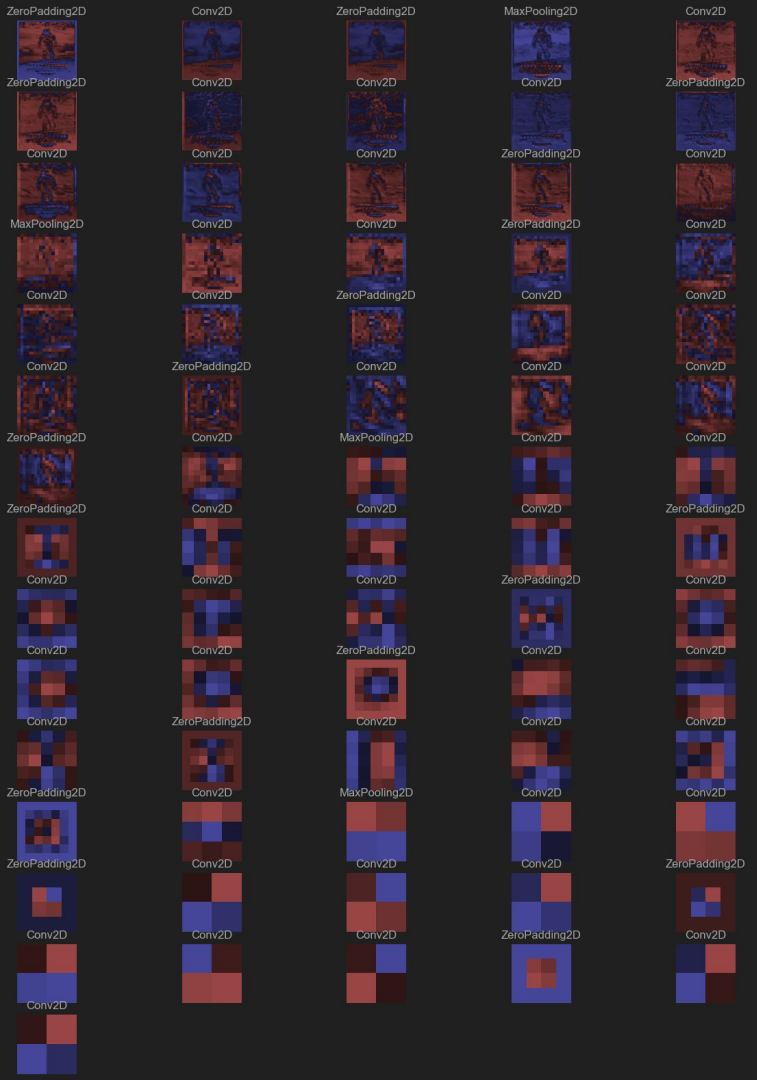
Model: Custom, Genre: Horror

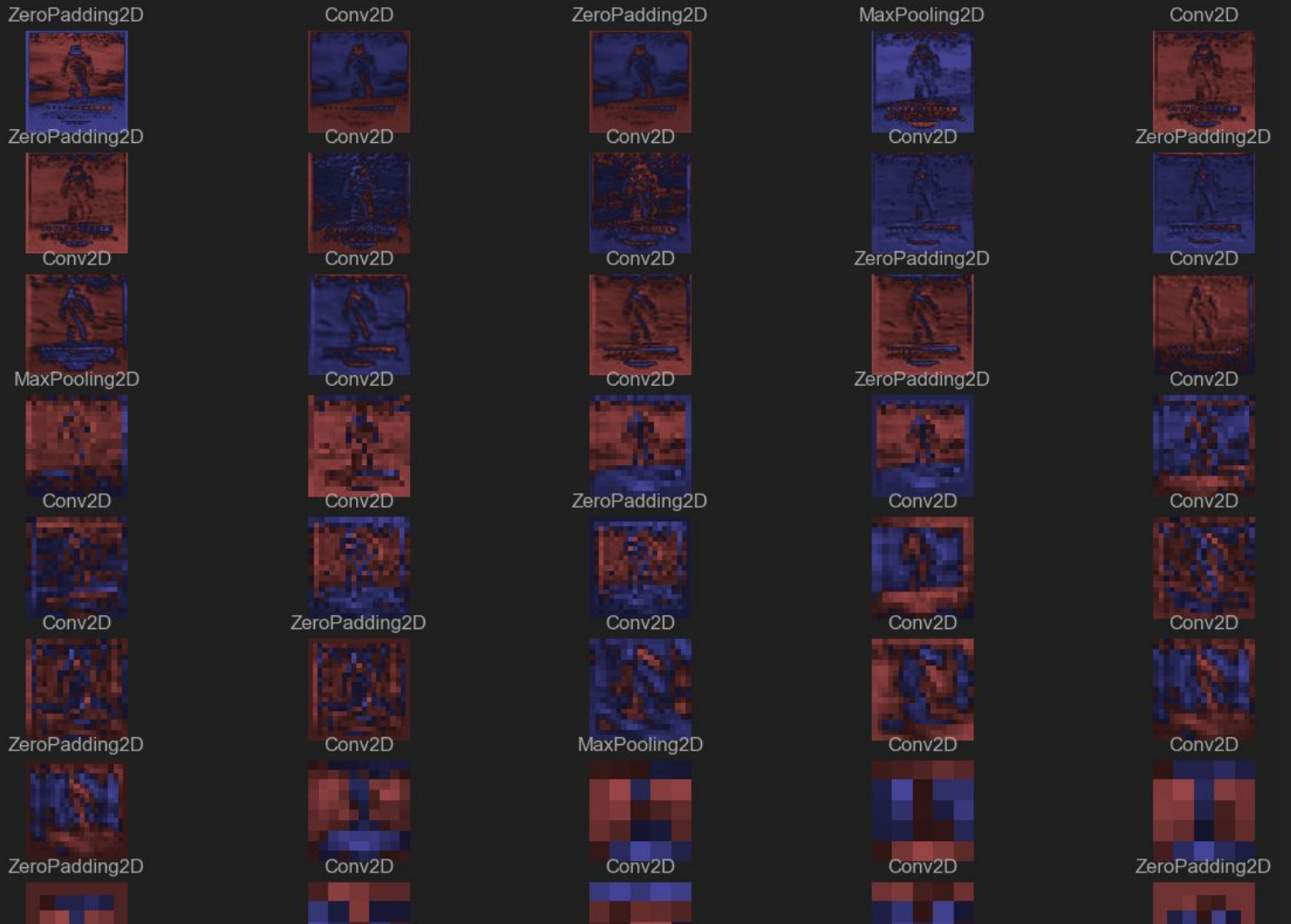


Model: Custom, Genre: Family



Resnet


ZeroPadding2D
Conv2D
ZeroPadding2D
MaxPooling2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
MaxPooling2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
MaxPooling2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
MaxPooling2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D


ZeroPadding2D
Conv2D
ZeroPadding2D
MaxPooling2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
MaxPooling2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
MaxPooling2D
Conv2D
Conv2D
ZeroPadding2D
Conv2D
Conv2D
Conv2D
ZeroPadding2D

Quellen

Abbildung 1: Filmposter
https://pngtree.com/freebackground/wall-of-movie-posters-on-display_3620105.html

Abbildung 2-5:
https://www.learndatasci.com/tutorials/hands-on-transfer-learning-keras/