

PROJECT WORK : IE 406 – MACHINE LEARNING

MACHINE LEARNING ALGORITHMS FOR MOVIE GENRE PREDICTION

GROUP : 26

Assigned By : Prof. Manjunath V. Joshi

Group Members:

- o Abhishek Sutariya – 201701012
- o Purva Singhvi – 201701039
- o Karpit Patel – 201701174
- o Dharmin Solanki – 201701198

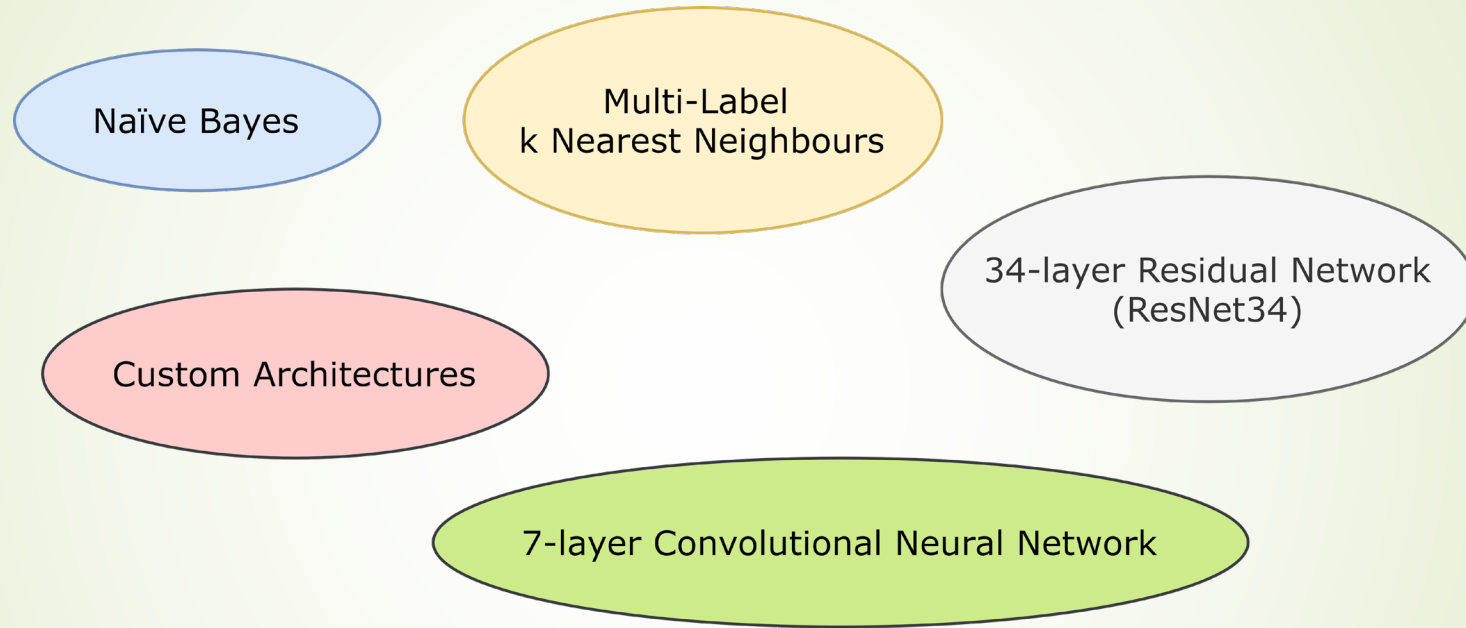
Problem Statement

Given a movie poster, the big picture is to predict the genre of the movie. The model should be able to learn and classify various features based on colors and contrasts as well as structural cues and perform feature extraction. The movie poster would be a color image and the output would be the genre.



Animation, Adventure

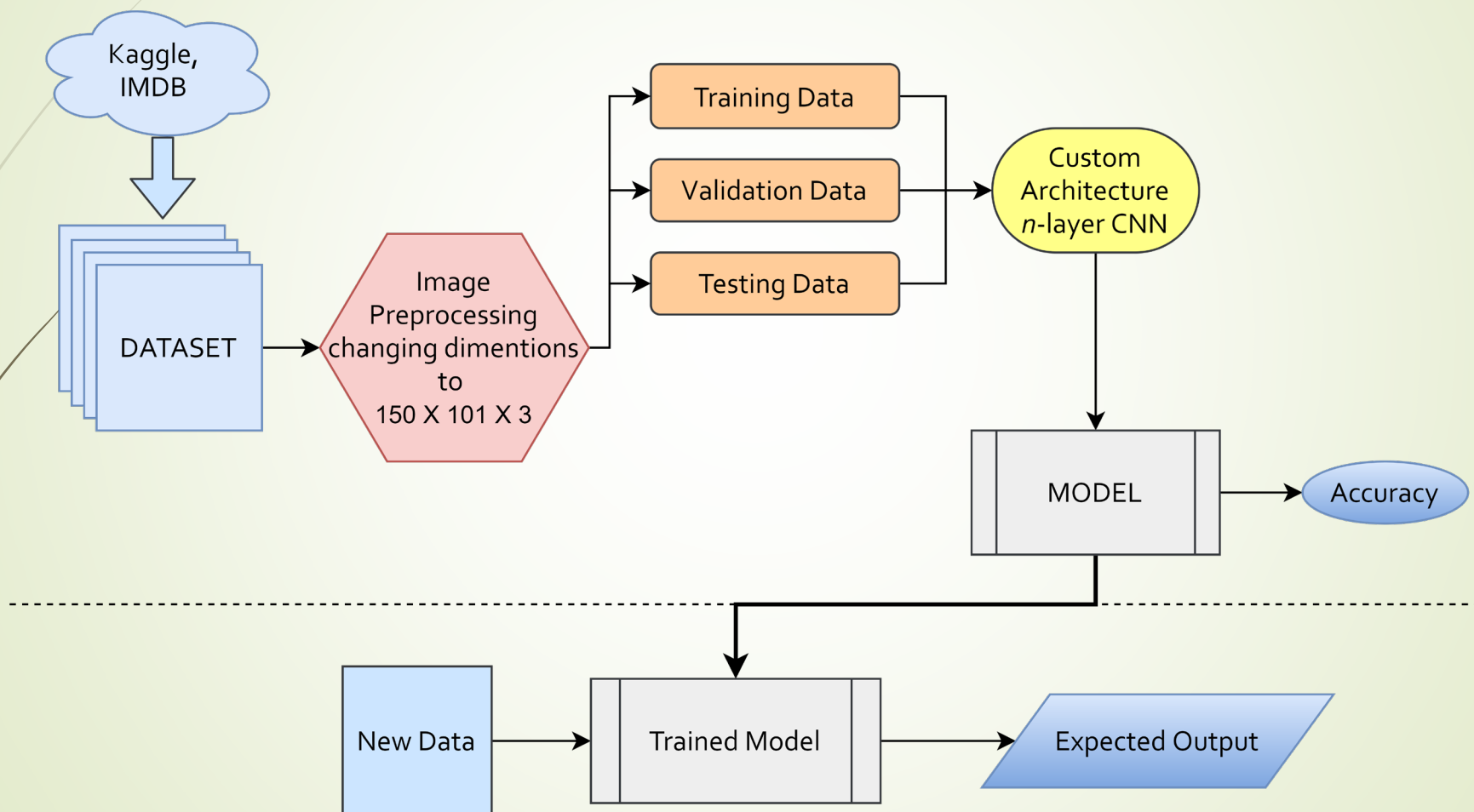
Previous Work Done



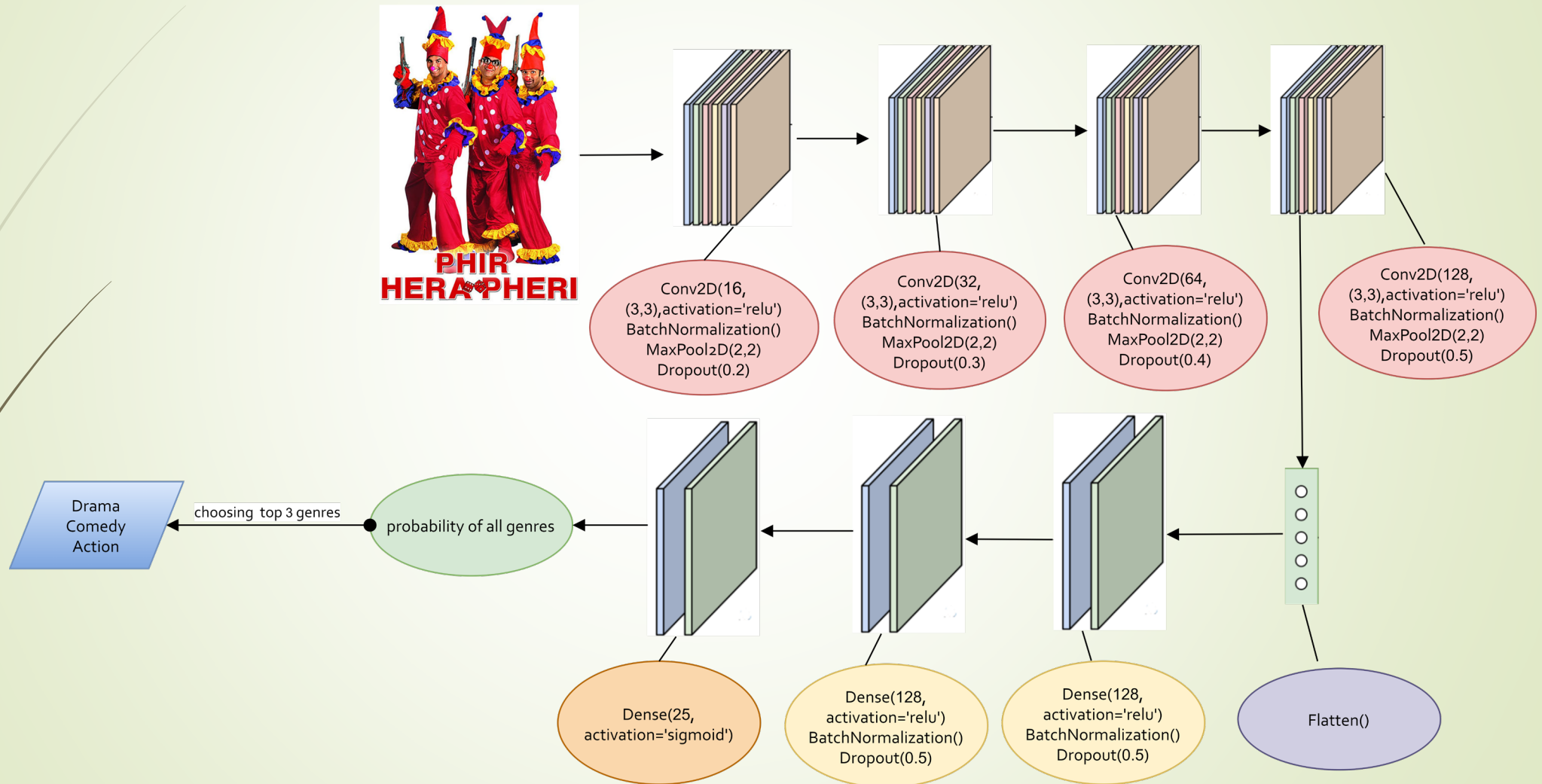
...and many!

4

Block Diagram



Workflow using an example



Code Snippets

Image Pre- Processing

```
img_width=101
img_height=150

x=[]

for i in tqdm(range(data.shape[0])):

    path = 'E:/Semester 6/IE 406 - 4.0 - Machine Learning/Project/codes/Images/' +data['Id'][i] + '.jpg'
    pic = image.load_img(path,target_size=(img_height, img_width, 3))
    pic = image.img_to_array(pic)
    pic = pic/255.0
    x.append(pic)

X=np.array(x)
```


Custom CNN Architecture

```
model = Sequential()

model.add(Conv2D(16, (3, 3), activation='relu', input_shape=X_train[0].shape))
model.add(BatchNormalization())
model.add(MaxPool2D(2, 2))
model.add(Dropout(0.2))

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

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

model.add(Conv2D(128, (3, 3), (activation = 'relu'))))
model.add(BatchNormalization())
model.add(MaxPool2D(2, 2))
model.add(Dropout(0.5))

model.add(Flatten())

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

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

model.add(Dense(25, (activation = 'sigmoid'))))
```



Testing

```
n_test = 100
X_test = X[n:n + n_test]
y_test = y[n:n + n_test]

pred = model.predict(np.array(X_test))
pred.shape
```

```
=====OUTPUT=====
( 100, 25 )
```



```
Accuracy

def accuracy_score(y_test, pred):
    value = 0

    for i in range(0, len(pred)):
        first3_index = np.argsort(pred[i])[-3:]
        correct = np.where(y_test[i] == 1)[0]
        flag=1

        for j in first3_index:
            if j in correct:
                if flag==1:
                    value += 1
                    flag=0

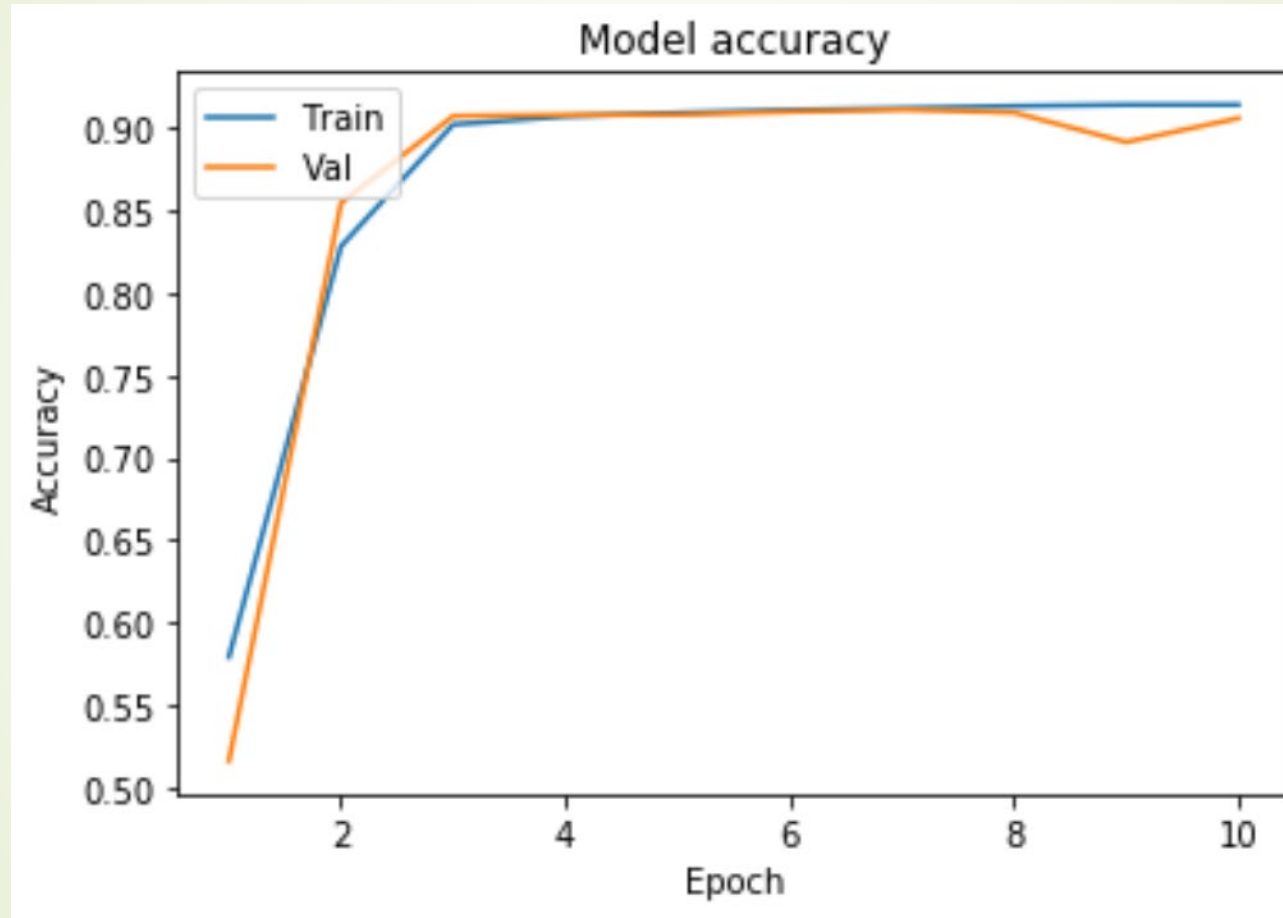
    acc = value/len(pred)

    return acc*100

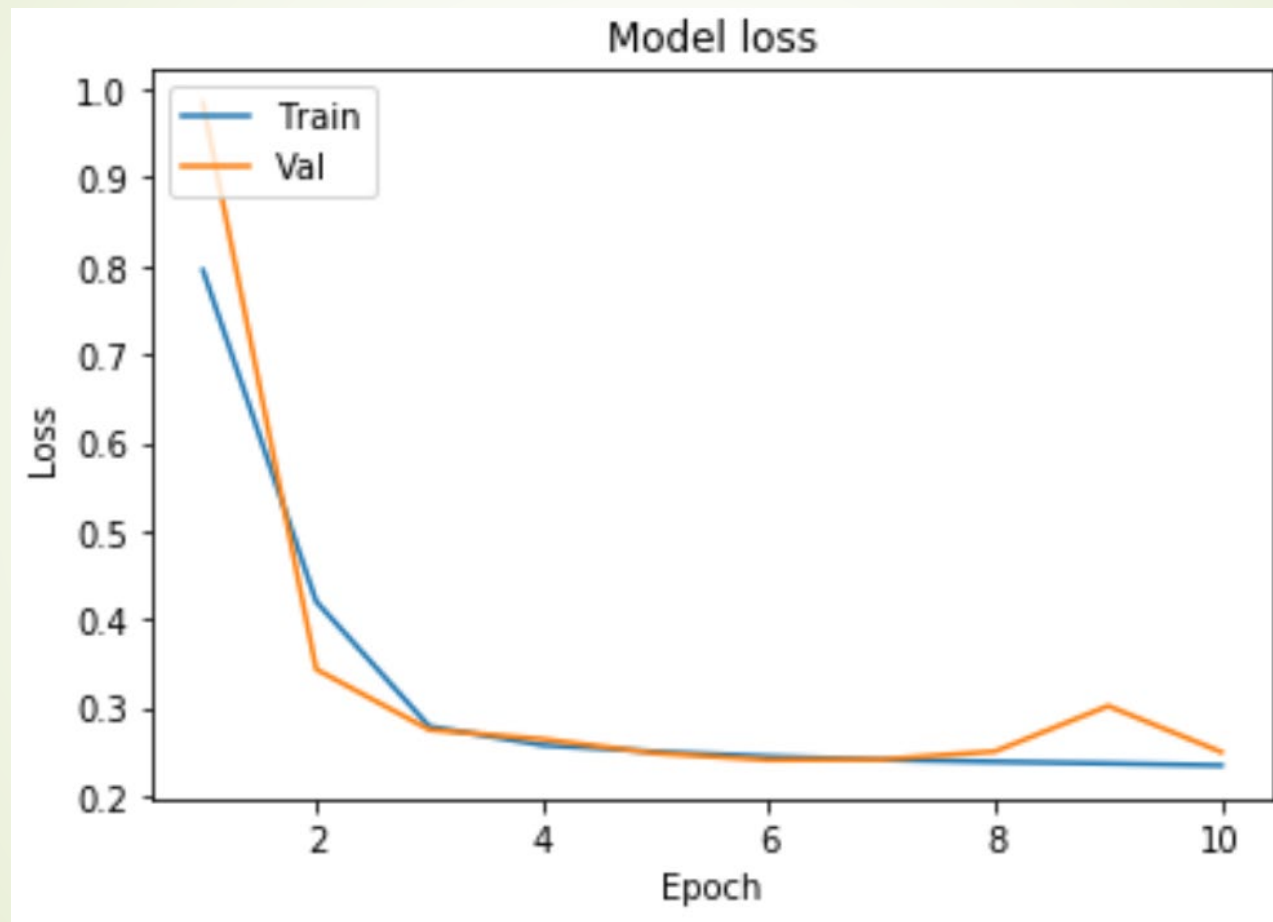
print("Accuracy =", accuracy_score(yTest, pred))

=====OUTPUT=====
Accuracy = 81.0 %
```

Results



Results



References

Data Links

- <https://www.kaggle.com/rounakbanik/the-movies-dataset/kernels>
- <https://www.kaggle.com/neha1703/movie-genre-from-its-poster>
- <https://www.imdb.com/>
- https://github.com/laxmimerit/Movies-Poster_Dataset

Reference Links

- <http://cs229.stanford.edu/proj2019spr/report/9.pdf>
- <http://cs229.stanford.edu/proj2019spr/poster/9.pdf>
- <https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624>
- <https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff>
- <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
- <https://towardsdatascience.com/an-introduction-to-the-naive-bayes-algorithm-be3bd692273e>
- <https://youtu.be/4HKqjENq9OU>
- <https://towardsdatascience.com/introduction-to-resnets-coa830a288a4>