cs577 Final Project: Report

Yuanxing Cheng, A20453410, CS577-f22
Department of Mathematics
Illinois Institute of Technology

November 16, 2022

1 Problem statement

We are interested in using movie posters to predict the movie genres. This is a multi-classification problem. And we belive the model would work better given extra object detection of input along with the original input. This idea I intend to implement is from [6].

We also noticed a related work using gram layer [10] to connect the poster and the genre. However since my teammate drop the course, this part cannot be completed.

More unfinished work can be found in the tesm responsibility section.

2 Proposed solution

I split the model into three part, which are base CNN layer, yolo objective detection layer, and gram layer. The data collection should be done by my teammate, and he quited. Thus I don't have access to the training data. The following model involving input shape and total number of movie genre are instead assumed.

The outputs of the three parts (now two parts) are then added together and passed to a full connected layer, then another full connected layer with softmax activation.

In the original paper, the authors adopted an algorithm that could obtain a probability thresholds from a discret probability distribution. So that we can use the obtained thresholds to decide if a movie is labeled as certain genre or not. I followed this idea and implement it. Here's the algorithm 1.

3 Implementation details

All the code are written using Python [9] and was going to be tested on colab platform. For now, I use the following modules to help me in construct the model: Numpy [7], keras [5] and tensorflow [1].

component: CNN

This is a normal CNN layer consists of an up sampling process and down sampling process

Algorithm 1 Algorithm for obtaining the probability thresholds

Input: model output, the predicted probability vector $\hat{\boldsymbol{y}}$, the true probability $\boldsymbol{y}$, obtained from dataset. For example, if movie is of genre 1 and 2, then this true probability vector is $(0.5, 0.5, 0, 0, \dots)$. An predetermined threshold upper bound u and threshold moving stepsize s. All vectors are of length equal to the total number of movie genre, N.
Output: vector of best threshold for current output $\boldsymbol{y}_s$
Initialize $\boldsymbol{y}_s$
for $i = 1$ to N **do**
  $j = 0$
  Initialize empty vector $\boldsymbol{\rho}$ and $\boldsymbol{\theta}$
  for $j < u$ **do**
    Zero initialize a binary vector $\boldsymbol{b}$
    if $\hat{\boldsymbol{y}}_i > j$ **then**
      $b_i = 1$
    else
      $b_i = 0$
    end if
    assign t to be the Matthews correlation coefficient of $\boldsymbol{y}$ and $\boldsymbol{b}$, obtained using [8]
    append $\boldsymbol{\rho}$ the value t
    append $\boldsymbol{\theta}$ the value j
    $j \leftarrow j + s$
  end for
  $k^* = \arg\max_k \boldsymbol{\rho}$
  $(\boldsymbol{y}_s)_i = \boldsymbol{\theta}[k^*]$
end for

Layer	output shape	params used
input layer	1120,960,3	0
conv 2d	1120,960,100	2800
max pooling	560,480,100	0
conv2d	560,480,32	28832
conv2d	560,480,64	18496
conv2d	560,480,128	73856
conv2d	560,480,256	295168
conv2d	560,480,128	295040
conv2d	560,480,80	92240
batch normalizing	560,480,80	320
max pooling	280, 240 ,80	0
flatten	5376000	0
dense	80	430080080
batch normalizing	80	320

3.1 component: object detection with Yolo

This part is adpoted from [3] and [4], where the first one gives an implementation of the original yolo model in keras, and the latter one give an thorough example which help me a lot in understanding this model.

Here is a figure 1 that explains its basic architecture from [2].

Above yolo detection base is then followed by certain up sampling and a dense layer. Overall, this part looks like the following 2.

3.2 component: gram layer

This part should be finished by my teammate and thus is skipped here.

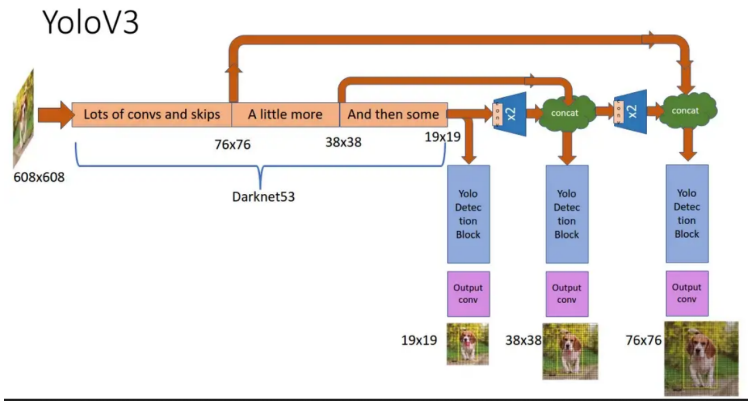

Figure 1: *yolov3 architecture from Uri Almog*

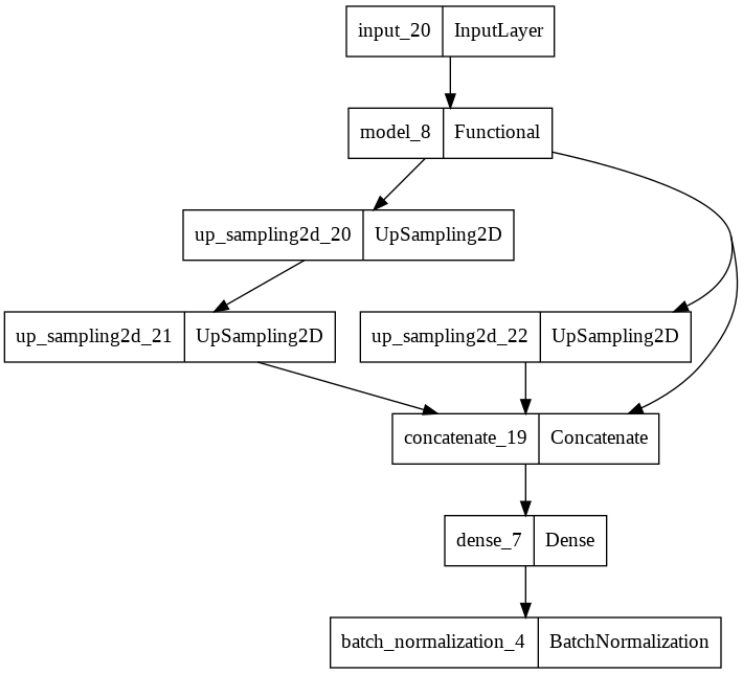

Figure 2: *full yolov3 branch*

This three components are designed to parallelly extract features from input. The following figure explain this architecture.

4 Results and discussion

Due to lack of input data, the model is not trained. I can only provide a synthetic data to find a best threshold.

5 Team responsibility

This section follows the original proposal before my teammate quit the course. As been decided, Yuanxing Cheng will be doing the implementation of [6] and the paper, slides. Shansi Dong will be doing the imple-

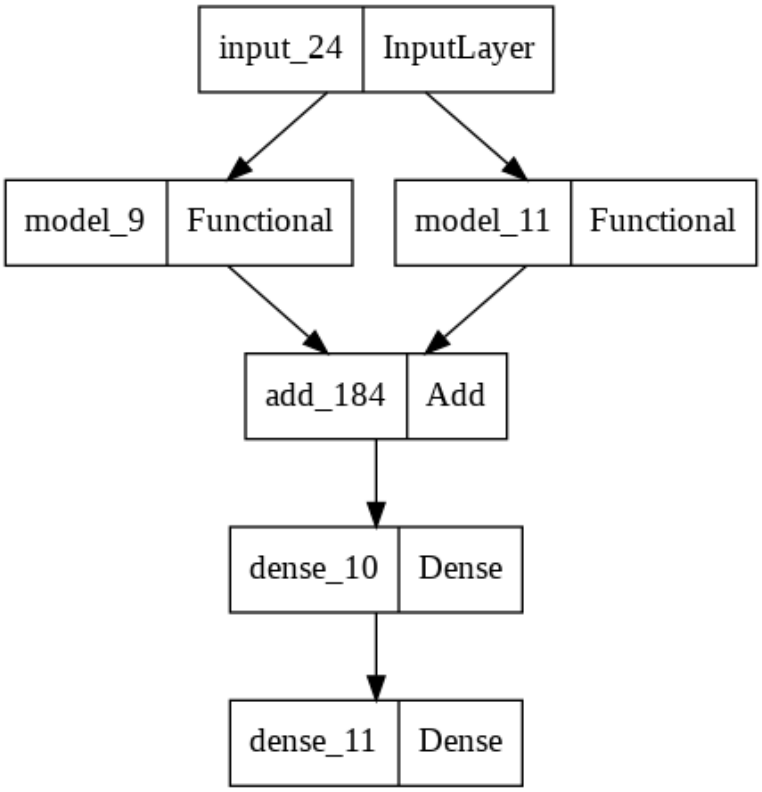

Figure 3: *overall model with 3 parallel components, notice the gram layer component is omitted here*

mentation of [10] and data collection. As he quit the course, the extra model compoment is not constructed in our final model. In addition, due to the fact that I have limited knowledge on collecting poster data from IMDB, the training and testing on the model cannot be completed.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Almog, U. YOLO V3 Explained.

[3] Anh, H. N. YOLO3 (Detection, Training, and Evaluation).

[4] Brownlee, J. How to Perform Object Detection With YOLOv3 in Keras.

[5] Chollet, F., et al. Keras. https://keras.io, 2015.

[6] Chu, W.-T., and Guo, H.-J. Movie genre classification based on poster images with deep neural networks. In *Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes* (2017), pp. 39–45.

[7] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature 585*, 7825 (Sept. 2020), 357–362.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[9] Van Rossum, G., and Drake, F. L. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[10] Wi, J. A., Jang, S., and Kim, Y. Poster-Based Multiple Movie Genre Classification Using Inter-Channel Features. 66615–66624.