Find Genre from Book Cover!!

Furkan Karadeli
Hacettepe University
Computer Science
furkankaradeli@hacettepe.edu.tr

Erhan Kabaoğlu
Hacettepe University
Computer Science
erhankabaoglu@hacettepe.edu.tr

Abstract

We read books for a very long time but it is not still enough understand book genre at a glance. For understand topic of book we have to read 1 or 2 pages or look at contents. Can we understand faster?

We used neural networks and text classification algorithm together but with 2 different ways. 1 way is both of networks has own loss function and own accuracy values. We add these values in last layer. However, second one is different from first one. We used text classification as a feature vector for neural networks.

1. Introduction

When we decide to buy book firstly we go the section what we love. When a seller wants sell a book he/she has to define the book's genre. In this article we will show you we can do it in faster way. And if it works on book it can be works on movie and album cover as well.

1.1. Problem Description

Our purpose is defining a books genre by get information from its cover. Specialy, given a book's cover and title, we will attempt to predict its genre.

1.2. Challenges

In this project there is a lot of variable which can be changeable such as batch size, learning rate, number of epochs, number of frozen layer and neural network type. For to overcome this variable we had to a lot of make process. And most important thing was should we use BOW algorithm as a feature vector or not?

2. Related Works

Actually this problem can solve by classic neural networks algorithm with machine learning. However, this type learning's cost is too much. Too much time and too much image for each class. So this way is not be effective even maybe it won't work for too many class. In one project[1] text classification and neural networks use together and they achieved %80 accuracy but it is not enough.

2.1 Image Classification

The first type of networks we examined were alexnet. The AlexNet proposed by Alex Krizhevsky in his work has eight layers including five convolutional layers followed by three fully connected layers. Some of the convolutional layers of the model are followed by max-pooling layers. As an activation function, the ReLU function is used by the network which shows improved performance over sigmoid and tanh functions. Accuracy of alexnet is %57[2] in this example. And we want to try it. Our second type of network is vgg. VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes [3]. Our next neural networks are ResNet18 and ResNet50 respectively. Residual Network (ResNet) is a Convolutional Neural Network (CNN) architecture which was designed to enable hundreds or thousands of convolutional layers. While previous CNN architectures had a drop off in the effectiveness of additional layers, ResNet can add a large number of layers with strong performance. And these architectures can be use classification. So we can use them to predict genre of books. And last neural network is ResNet152. This architecture has 152 layers. For this article[6] ResNet152 has %19.38 top-1 error, so this mean it is better than other ResNet architectures.

2.2 Text Classification

In this paper we are going to show that how we ensamble image classification and text classification together and use bag of words algorithm as a feature vector. In this article[4] shown that it is possible to achieve high accuracy using pretrained vectors to represent features of different words in a CNN architecture. So we think we can ensemble 2 classification model. On the other hand we can use BOW algorithm as a feature vector. For this article [5] we can achieve %80 accuracy with combine bag of words and image classification.

3. Models

We use following models in our experiments. We use pretrained models because of lack of data. We want to see results of different models to select best one.

3.1 Alexnet

Alexnet is the first CNN architecture. Firstly we examined all layers. Then we try freeze different layers and freeze first 2 block was the best choose for integrated methods. We freeze 1 block of the model to avoid underfitting in just image classification method.

3.2 VGG16

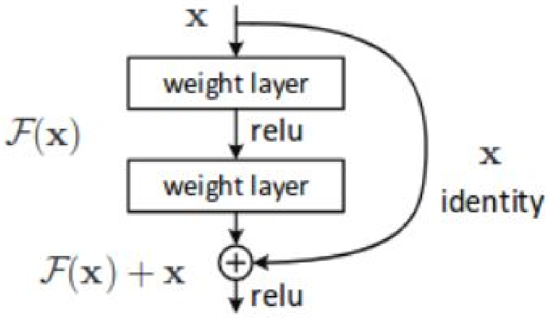
VGG-16 architecture has 16 layers they are 13 convolutional layers and 3 fully connected layers. Like in Alexnet we firstly examined all layers but it was not effective and it takes too much time and we are face with overfitting problem. So we decide freeze some of layers. We freeze all conv layers for integrated methods. We freeze most of conv layers for just image classification method. And finally we got better results.

3.3 ResNet18

Resnet18 architecture is the one of the most used in image classification algorithm and consist of residual modules. We freeze all conv layers for integrated methods. We freeze most of conv layers for just image classification method.

3.4 ResNet50

Each 2-layer block is replaced in the 34-layer net with this 3-layer bottleneck block, resulting in a 50-layer ResNet (see above table). They use option 2 for increasing dimensions. This model has 3.8 billion FLOPs. For example residual networks works like this:


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

3.5 ResNet152

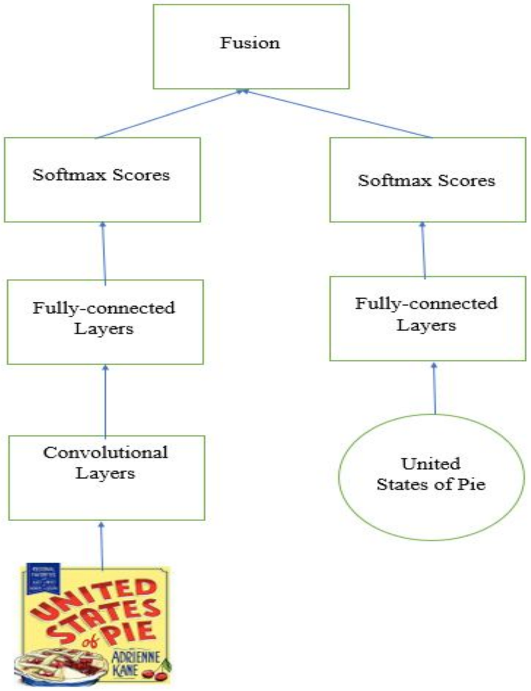
This architecture more advanced version of resnet50 and resnet18. This architecture has exactly 152 layers. We think that this model can better results because it uses more pretrained conv layers and it may help to extract feature from book covers. It takes too long time, but it has the best error rate among our models[6].

4. Methods

We use 2 different methods to classify book genre.
We think that using only image classification or bow classification is not enough. So we combine two approach in different ways.

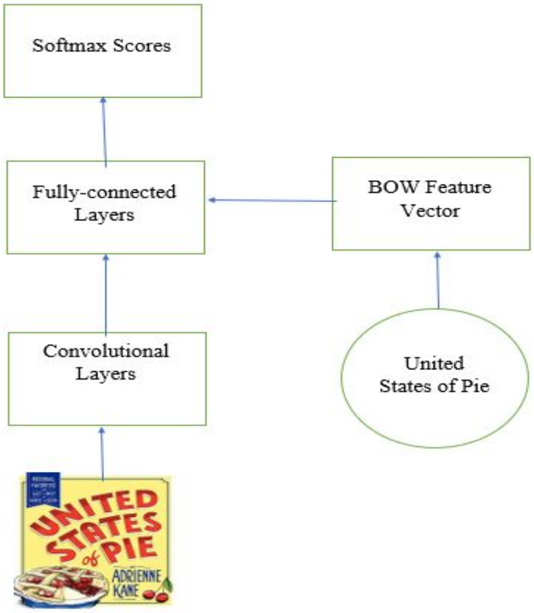
4.1 Fusion Convolutional Neural Network and Bow Classifier

In this method we initialize one of the image classification model and bow classification model. We combine softmax scores and update parameters together. Here is illustration:



4.2 Convolutional Neural Network with Bow Feature Vector

In this method, We use bag of words algorithm again and used them as a feature vector. We concatenate bow feature vector to the first fc layer in all models. Here is illustration:

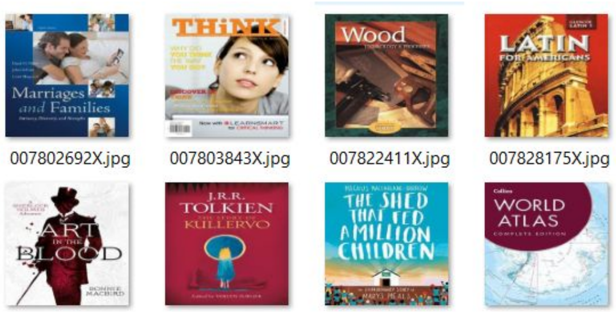


4.3 Convolutional Neural Network

We will use different CNN architectures to classify images. Main purpose of this method is that show weakness of using just Convolutional Neural Network across to ensemble methods.

5. Dataset

We will use 28,500 book covers and titles and 15 categories collected from Amazon.com. This dataset is adapted from BookCover30 used by Iwana and Uchida 2016[7]. As a total 25,650 training dataset and 2850 test dataset. We refer to use 15 genres and they are Arts & Photography, Biographies & Memoirs, Business & Money, Calendars, Children's Books , Comics & Graphic Novels, Computers & Technology, Cookbooks, Food & Wine, Crafts, Hobbies & Home, Christian Books & Bibles, Engineering & Transportation, Health, Fitness & Dieting, History, Humor & Entertainment and Law. Each genre has 1539 training samples, 171 validation samples and 190 test samples.



6. Experimental Settings

We try different hyperparameters like that batch size, number of freezing layers, learning rate, drop-out, optimizer, regularization. It is a challenging problem. We get help from Hyper-Parameter Optimization: A Review of Algorithms and Applications written by Tong Yu and Hong Zhu[8].

6.1 Preprocessing

We resize images into 224x224 and save them into HDF5 files for computation efficiency.

For image train dataset, we apply random horizontal flip with p = 0.5 probability for data augmentation. We normalize our images with mean 0.485 0.456 0.406 and with standard deviation 0.229, 0.224, 0.225[9].

For image validation and test datasets, we just normalize images at the same values with train dataset.

For text dataset, we tokenize titles which means, this breaks up the strings into a list of words or pieces based on a specified pattern using regular expressions. We convert all letters to lowercase and remove punctitions. We remove not alphabetic characters and stop words because they have low predictive power and can hurt accuracy. Finally, we divide the words to their roots (stemming) because conjugated words cause redundancy and they can loss actual meanings.

After cleaning data, we create vocabulary using words, we keep these into dictionary with corresponding index numbers.

After all, we create bow vectors using vocabulary indexes. we choose word frequency representation for bow feature vectors[10].

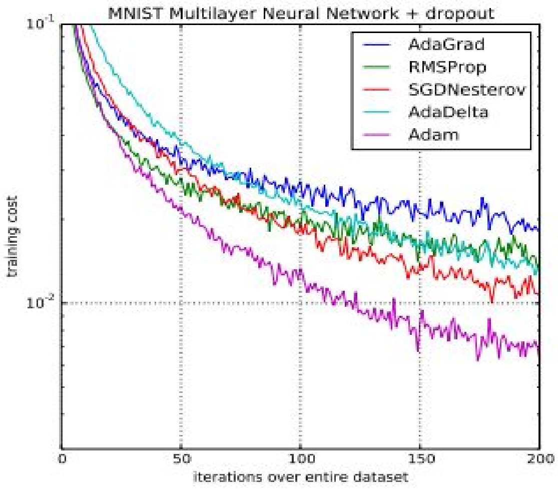
6.2 Training

As you can see above, we try different hyperparameters and choose the best ones. we give you information about last version of hyperparameters.

We train our all models in 25 epochs because almost all models reach close to 100 train accuracy value.

We set batch size to 16 with a single GPU with shuffling data.

We use Adam optimizer with default parameters learning rate(alpha) 0.001 , beta1 = 0.9, beta2 = 0.999[11].



It's calculate exponential moving average(EMA) to reach optimum values of weights and this speed up performance. You can see an example of comparison of different optimizer algorithms above.

We use initial learning rate = 0.001 but our model had trouble finding optimum after a while. So, we use Step Learning Scheduler with gamma = 0.1, step_size=10. This help to reach local(global) optimum.

We use Cross Entropy Loss function, its convenient for classification tasks because its uses softmax function implicitly and give output that probability value between 0-1 [12].

Cross Entropy Loss function calculated as:

$$-(y\log(p) + (1-y)\log(1-p))$$

$$-\sum_{c=1}^{M} y_{o,c}\log(p_{o,c})$$

M = number of classes
y = binary indicator (0 or 1)
p = predicted probability observation o is of class c

When training we check validation accuracy for per epoch and we keep track of best validation accuracy value and save them after training. Also we calculate top1, top2 and top3 accuracies with dividing correct predictions to total size of dataset.

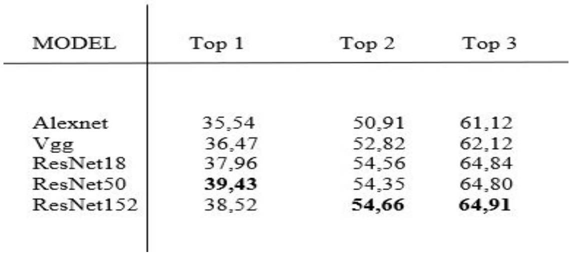
We train 3 methods which are just bow classification, just CNN image classification, fusion of bow classification and CNN image classification, concatenate bow feature vector to CNN fully-connected layer and all configurations same for every method except freezing layers.

6. Experimental Results

In this part we are going to show you our experimental results with table and graphic.

6.1 Convolutional Neural Network

Our first experiment like we wrote that was just using neural networks. We have 5 neural networks and this table include these top 1, top 2 and top 3 accuracy values.



MODEL	Top 1	Top 2	Top 3
Alexnet	35,54	50,91	61,12
Vgg	36,47	52,82	62,12
ResNet18	37,96	54,56	64,84
ResNet50	**39,43**	54,35	64,80
ResNet152	38,52	**54,66**	**64,91**

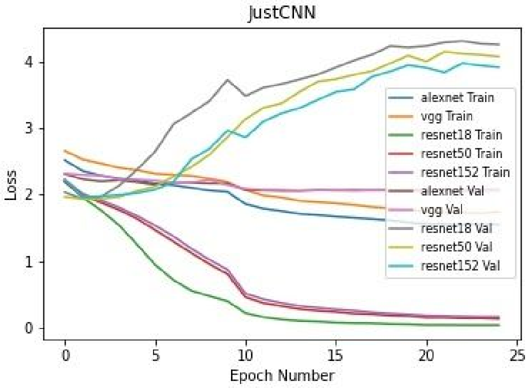


Figure 1. Loss values for train and validation (Just CNN)

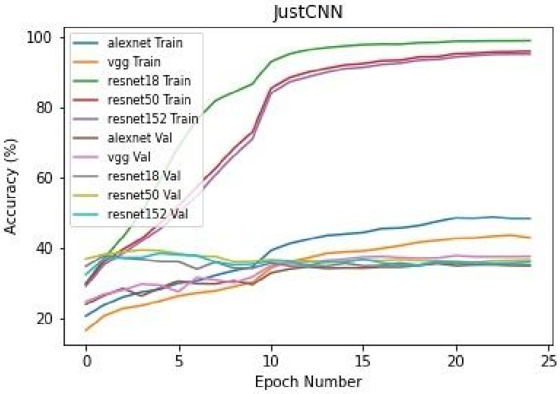


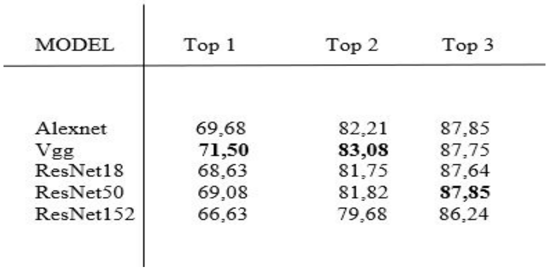
Figure 2. Accuracy values for train and validation (Just CNN)

Like we can see just using convolutional neural network for classification in this task is not useful method. Even ResNet50 had best accuracy, its top 1 accuracy is 39,43. To conclusion, using just neural network is not helpful about genre classification.

6.2 Fusion Convolutional Neural Network and Bow Classifier

In this experiment we combine 2 softmax score. One of them is coming from image classification and other is coming from bow classification. In first figure we can see train and accuracy loss values for each neural network. In table 1 we see top the results fusion of neural networks and bow classifier. As we expected accuracy values increase while uses of deeper networks.Vgg has achieving best accuracy value. In figure 1 we see that loss values of train and validation steps. Loss values decrease until 0.60. In figure 2 we can see that accuracy values. Validation accuracy value increase at each epoch.

Table1. Results from cnn with bow vector



MODEL	Top 1	Top 2	Top 3
Alexnet	69,68	82,21	87,85
Vgg	**71,50**	**83,08**	87,75
ResNet18	68,63	81,75	87,64
ResNet50	69,08	81,82	**87,85**
ResNet152	66,63	79,68	86,24

Figure 1. Loss values for train and validation steps (Fusion)

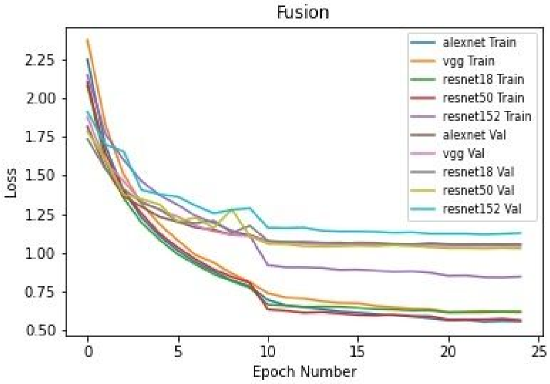


Figure 1. Loss values for train and validation (Fusion)

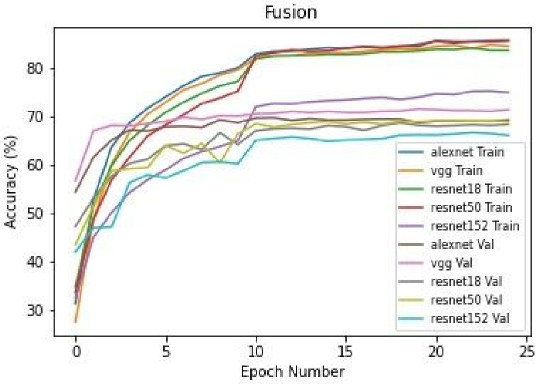


Figure 2. Accuracy values for train and validation (Fusion)

6.3 Convolutional Neural Network with Bow Feature Vector

In table 1 we see top the results of neural networks with bow feature vector. As we expected accuracy values increase while uses of deeper networks. ResNet50 has has achieving best accuracy value. In figure 1 we see that loss values of train and validation steps. Loss values decrease in first 15 epoch then it stops same value.In figure 2 we can see that accuracy values. Validation accuracy value increase at each epoch. Examining the confusion matrix in figure 3 we see that “Calender” genre is the most predictable class. However, our model has a lot of mistakes between “Biographies & Memories” and “History” class because of they are similar class.

Table1. Results from cnn with bow vector

MODEL	Top 1	Top 2	Top 3
Alexnet	35,75	50,66	60,45
Vgg	38,14	52,56	62,91
ResNet18	69,05	81,68	**87,85**
ResNet50	**69,92**	81,75	87,68
ResNet152	69,50	**81,92**	86,98

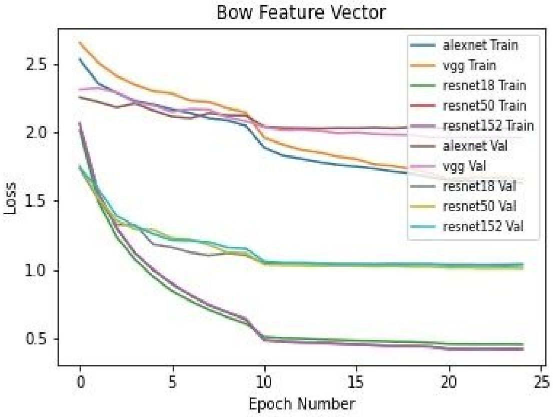


Figure 1. Loss values for train and validation (Bow feature vector)

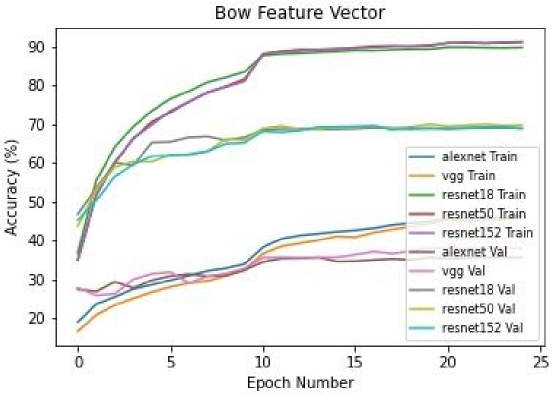
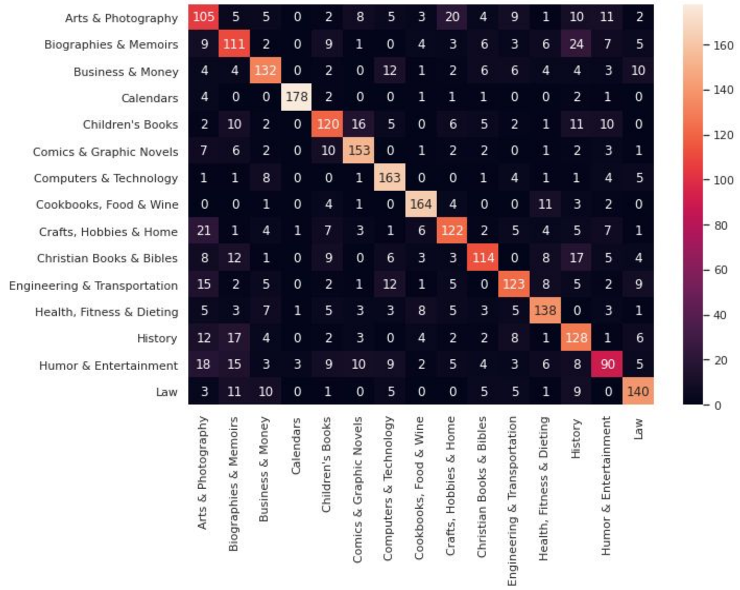


Figure 2. Accuracy values for train and validation (Bow feature vector)



	Arts & Photography	Biographies & Memoirs	Business & Money	Calendars	Children's Books	Comics & Graphic Novels	Computers & Technology	Cookbooks, Food & Wine	Crafts, Hobbies & Home	Christian Books & Bibles	Engineering & Transportation	Health, Fitness & Dieting	History	Humor & Entertainment	Law
Arts & Photography	105	5	5	0	2	8	5	3	20	4	9	1	10	11	2
Biographies & Memoirs	9	111	2	0	9	1	0	4	3	6	3	6	24	7	5
Business & Money	4	4	132	0	2	0	12	1	2	6	6	4	4	3	10
Calendars	4	0	0	178	2	0	0	1	1	1	0	0	2	1	0
Children's Books	2	10	2	0	120	16	5	0	6	5	2	1	11	10	0
Comics & Graphic Novels	7	6	2	0	10	153	0	1	2	2	0	1	2	3	1
Computers & Technology	1	1	8	0	0	1	163	0	0	1	4	1	1	4	5
Cookbooks, Food & Wine	0	0	1	0	4	1	0	164	4	0	0	11	3	2	0
Crafts, Hobbies & Home	21	1	4	1	7	3	1	6	122	2	5	4	5	7	1
Christian Books & Bibles	8	12	1	0	9	0	6	3	3	114	0	8	17	5	4
Engineering & Transportation	15	2	5	0	2	1	12	1	5	0	123	8	5	2	9
Health, Fitness & Dieting	5	3	7	1	5	3	3	8	5	3	5	138	0	3	1
History	12	17	4	0	2	3	0	4	2	2	8	1	128	1	6
Humor & Entertainment	18	15	3	3	9	10	9	2	5	4	3	6	8	90	5
Law	3	11	10	0	1	0	5	0	0	5	5	1	9	0	140

Figure 3. Confusion matrix for the ResNet152 and BOW feature vector

7. Discussion and Conclusions

This paper shows us we can classified books by genre from its cover and its title. Our model is successful because we use deeper networks with words. We get %70 accuracy for top 1, %81,92 accuracy top 2 and %86,98 accuracy for top 3. In this project we tried different parameters like number of epoch, learning rate, batch size and optimizer type. But finally we get best parameters from our experiments. For a better accuracy we can use different text classification algorithm such as SVM, KNN etc [13]. Or if we use more datas with books cover and title we can get better results. Also some books can belong to different category so we can use multi task learning to predict book genres.

References

[1] B. K. Iwana and S. Uchida. Judging a book by its cover. 2016

[2] Forrest N. Iandola , Song Han, SQUEEZENET: ALEXNET-LEVEL ACCURACY WITH 50X FEWER PARAMETERS AND <0.5MB MODEL SIZE, 2016

[3] VGG16 – Convolutional Network for Classification and Detection, https://neurohive.io/en/popular-networks/vgg16/

[4] Y. Kim. Convolutional neural networks for sentence classification. CoRR, abs/1408.5882, 2014.

[5] Sebastian Sierra, Fabio A. González, Combining textual and visual representations for multimodal author profiling, 2018

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 2016

[7] Iwana and Uchida, Book Cover Dataset, https://github.com/uchidalab/book-dataset , 2016.

[8] Tong Yu and Hong Zhu, A Review of Algorithms and Applications, 2020

[9] Understanding transform.Normalize( ), 2018

[10] Rong Jin and Zhi-Hua Zhou, Understanding bag-of-words model: A statistical framework , 2010

[11] Diederik P. Kingma and Jimmy Lei Ba, ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION, 2015

[12] Zhilu Zang and Mert R. Sabuncu, Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels, 2018

[13] Rajni Jindal, Ruchika Malhotra and Abha Jain, Techniques for text classification: Literature review and current trends, 2015