BEN-GURION UNIVERSITY
OF THE NEGEV

FACULTY OF ENGINEERING SCIENCE

DEPARTMENT OF SOFTWARE AND INFORMATION SYSTEMS ENGINEERING

ADVANCED TOPICS IN RECOMMENDER SYSTEMS COURSE

Visually Aware Recommender System

2021

IMPROVING PAPER: VISUALLY-AWARE FASHION RECOMMENDATION AND DESIGN WITH GENERATIVE IMAGE MODELS

For Github Click Here

Submitted by

Amir Gabay, ID: 205381684
gabayam@post.bgu.ac.il

Amit Damri, ID: 312199698
amirdamr@post.bgu.ac.il

Contents

1 Abstract

Building recommender systems for domains like fashion hides unique challenges due to the huge number of items and the recent changes of the items' styles and the users' preferences. Visually aware recommender systems address these challenges by incorporating visual signals in the recommendation system. These signals can improve the recommendation system's performances significantly by incorporating pre-trained models and fine-tune them by the training process of the recommender system. By doing so, we were able to utilize the high level characteristics of the pixel-level items' images in the recommendation system. Our contribution integrated Siamese Networks that are used to compare two different objects. By using this kind of models, we were able to address this task as an implicit feedback training process by a BPR optimization technique. We tested the integration of two different types of models: Convolutional Autoencoders (CAE) and pre-trained ImageNet models such as InceptionV3. The first one models the items' picture by their visual patterns, while the second one models the item as being related to a certain category of items (e.g. shirts, pants). We compare these two techniques and try to improve the recommendation system's performances in the paper of Kang, Fang, Wang, and McAuley [5].

2 Introduction

The main goal of a recommender system is to provide personalized recommendations to users [5]. One of the most important features it should take in account when doing so is the items in the system and whether the user showed positive feedback (e.g. clicks, purchases) on them or not. Traditional models take into account patterns of behaviors of the users such as clicks, purchases or ratings given and through that they predict the future behavior of the users.

In domains like fashion, the prediction task can be very challenging for two main reasons: the amount of items is very big and new items are being added consistently, and users' preferences and products' styles change over time, thus it's very hard to determine what is 'fashionable'. Visually-aware recommender systems address these challenges by incorporating the visual signals directly into the recommendation model. For doing so, there are various methods that mostly include Convolutional Neural Networks (CNNs) for the visual inputs processing.

One way for incorporating the visual inputs in the model are by pre-trained models. Such models are usually trained already on a dataset of images for a classification task. These models are good at describing high-level characteristics of an image (e.g. color, texture, shape), but not necessarily the subtle characteristics of 'fashion'. Nevertheless, these methods are able to give a good solution, especially in 'cold-start' settings where high-level image characteristics can be informative.

Another way to incorporate visual inputs are by using CNNs in various architectures directly in the model. These models are better in the specific task, but they require bigger datasets to learn from.

Convolutional autoencoder (CAE) models has shown good results in the task of embedding creation [7]. These models are able to create a limited sized vector that represents the image by rebuilding it. Recommender systems can take advantage of this type of models and use the embedding they create and incorporate them in the recommender system.

Siamese networks are known as a way to conduct comparative judgements on two different objects by using the exact same weights and model architectures on the two different objects [6]. In their paper, Kang et al. [5] use a siamese network to conduct comparative judgements between items' images. They have done so by creating vectors for each of the pair of items, and then they conduct an L1 distance to measure their distance. Then they used the Bayesian Personalized Ranking (BPR) [9] to optimize the implicit feedback based recommendation.

On this study we build upon the DVBPR method of Kang et al. [5] and we try to improve their siamese network that use simple CNNs by two different techniques: by a convolutional autoencoder based solution and by an ImageNet pre-trained InceptionV3 [12] model.

We perform experiments on three popular recommendation corpuses of Amazon and Tradesy that include hundreds of thousands users, items and reviews. The experiments were conducted on the two suggested improvements to the recommender system of Kang et al. [5]- the CAE based solution and the ImageNet based solution.

3 Background

3.1 Recommender Systems

Nowadays, online users are overloaded with information, so one of the industries' main tasks is to help them make better choices [10]. To do so, they use Recommender Systems. These systems seek to model compatibility between users and items, based on historical observations of clicks, purchases, or interactions [5]. The suggestions provided are aimed at supporting their users in various decision-making processes, such as what items to buy, what music to listen to, or what news to read [10].

One technique of these systems is Collaborative filtering (CF). These methods produce user specific recommendations of items based on patterns of ratings or usage (e.g., purchases) without a need for exogenous information about either items or users [2, 4, 10]. In order to relate between users and items, latent factor models, such as Matrix Factorization, can be used [2, 5, 10].

Matrix Factorization (MF) methods relate users and items by uncovering latent dimensions such that users have similar representations to items they rate highly [5, 10] and are the basis of many state-of-the-art approaches [5].

3.2 Visually Aware Recommender Systems

A variety of approaches have sought to incorporate deep learning into recommender systems [1, 3]. Visually aware Recommender Systems are one type of deep learning-based recommender system, where users' rating dimensions are modeled in terms of visual signals in the system (product images) [2, 4, 5]. In addition, modeling fashion or style characteristics has emerged as a popular computer vision task in settings other than recommendation [5], e.g., with a goal to categorize or extract features from images, without necessarily building any model of a user. This includes categorizing images as belonging to a certain style [2], as well as assessing items for compatibility [2].

3.3 Bayesian Personalized Ranking

Bayesian Personalized Ranking (BPR) is a state-of-the-art ranking optimization framework for implicit feedback [5, 9]. In BPR, the main idea is to optimize rankings by considering triplets

$(u, i, j) \in D$, where

$$D = \{(u, i, j) | u \in U \wedge i \in I_u^+ \wedge j \in I \backslash I_u^+\}.$$

Here $i \in I_u^+$ is an item about which the user u has expressed interest, whereas $j \in I \backslash I_u^+$ is one about which they have not. Thus, intuitively, for a user u, the predictor should assign a larger preference score to item i than item j. Hence, BPR defines

$$x_{u,i,j} = x_{u,i} - x_{u,j}$$

as the difference between preference scores, and seeks to optimize an objective function given by

$$\max \sum_{(u,i,j) \in D} \ln \sigma(x_{u,i,j}) - \lambda \|\Theta\|^2$$

where $\sigma(\cdot)$ is the sigmoid function, Θ includes all model parameters, and λ_Θ is a regularization hyperparameter. By considering many samples of non-observed items $j \in I \backslash I_u^+$, this method can be shown to approximately optimize the AUC in terms of ranking observed feedback for each user [5].

3.4 Siamese Networks

A Siamese neural network consists of twin networks which accept distinct inputs but are joined by a distance function at the top [4, 6]. This function computes some metric between the highest-level feature representation on each side [6]. The parameters between the twin networks are tied [6]. Weight tying guarantees that two extremely similar images could not possibly be mapped by their respective networks to very different locations in feature space because each network computes the same function [6]. Such models have been used to learn notions of style [4], including for fashion, by modeling the notion of item compatibility.

3.5 ImageNet Models

Traditionally, training of CNN models requires learning millions of parameters and requires a very large number of annotated image samples. This requirement prevents the application of CNNs to tasks with limited training data. Nowadays, there exist many pretrained CNNs which were trained on large scale data for different tasks and can be used in other tasks. This method of using pre-trained models is referred to as transfer learning. Transfer learning aims to transfer knowledge between related source and target domains. In this paper, we will use InceptionV3 pre-trained network.

The Inception model (GoogLeNet architecture) suggested by Szegedy et al. [11] introduced a new architecture and techniques that increased the performance on the ImageNet classification task. This model utilizes a new type of network-in-network blocks as a part of the network's architecture to increase the representational power of neural networks. A subsequent improvement to the Inception model was represented in [12] with the InceptionV3 model. This model used a new architecture which includes convolutional layers and Inception blocks that are different from the original GoogLeNet architecture. This model is much cheaper computationally and outperforms the results reported by [11].

3.6 Convolutional Autoencoders

Convolutional Auto-Encoders (CAE) combine the concepts of Convolutional Neural Networks and Auto Encoders. They use convolutional layers as part of the encoding-decoding process, and their goal is to reconstruct the input image [4]. They can extract data-driven features directly from three-dimensional maps, making them optimal for image processing [8].

4 Related Work

Visually aware recommender systems that use the BPR metric has shown to be successful on visual recommendation tasks. Therefore, we chose two methods that are visually aware based and that use the BPR metric. We will put our focus on both methods and will suggest an improvement based on them.

The first research was performed by Kang et al. [5]. They developed an end-to-end visually aware deep Bayesian personalized ranking method (called DVBPR) to simultaneously extract task-guided visual features and to learn user latent factors. Their goal was to generate for each user a personalized ranking over items with which the user has not yet interacted. Their idea was to train a CNN to extract visual features directly from the images, and simultaneously find a latent visual preference vector for each user. Finally, the predicted rating was a multiplication between both vectors. To optimize that problem and to find the right latent vectors, they used a BPR optimization function. In our research, we want to suggest two advantages. The first one is efficiency – we separate the training of the user and the items, and thus we can use pre-trained models what make the training easier. The second one is performance – we assume that the use of a pre-trained network will make it much easier to extract relevant features from the images.

The second research was performed by Hiriyannaiah et al. [4]. They used Convolutional Autoencoder (CAE) in order to extract visual features from images, and then they calculated the similarity between the trained feature vectors and the target feature vector in order to find similar items. They reported that their method got state-of-the-art results and that the CAE model succeeded in extracting useful visual latent features. Therefore, we decided to use this CAE model as a pre-phase of our method and train this model separately in order to get useful features that later can be used for items recommendation.

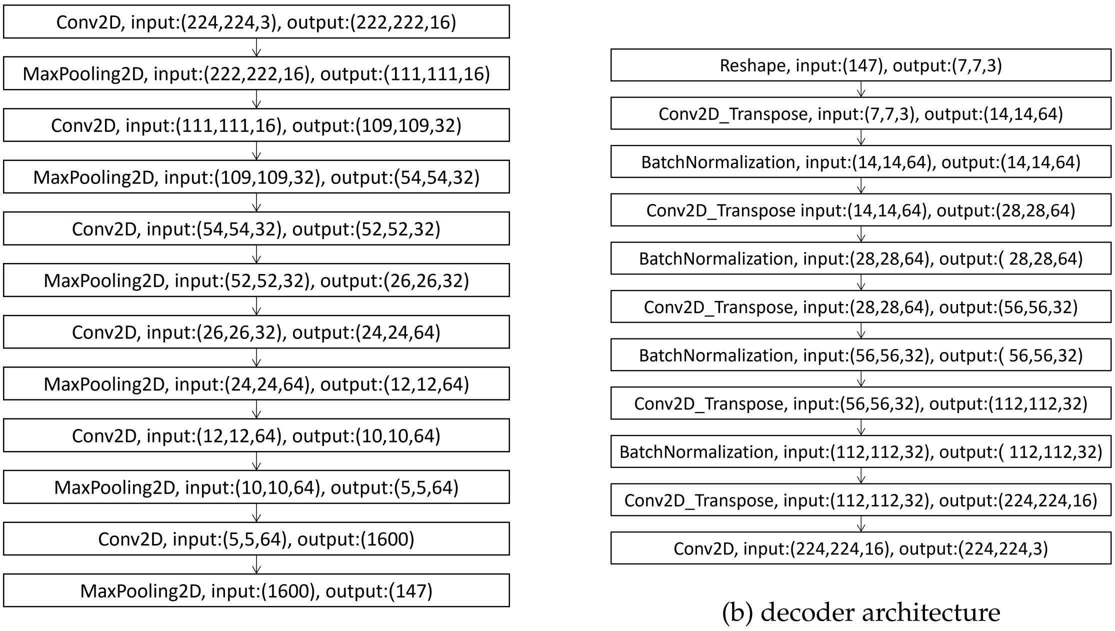
5 Our Method

Our method improves the visual component of the visually-aware deep Bayesian personalized ranking method (called DVBPR) that was developed by Kang et al. [5]. In the DVBPR model, they use a Convolutional Siamese Network to get a vector in size K to represent each image of an item. Our method improves this part of representing the item's image. Two different CNN architectures were applied in the architecture of [5].

5.1 Convolutional Autoencoder Based Solution

At first, an architecture of a convolutional autoencoder was integrated in the siamese network. The convolutional autoencoder tends to represent the images by visual features that are in the pixel level. Our intuition in this approach is that this way we can learn visual features that a user would like in different items, e.g. if a user likes items in a certain pattern like a floral pattern.

In this approach, the convolutional autoencoder is first trained on the items' images and then the encoder is used in the siamese network. The autoencoder's architecture is presented in figure 5.1. When the encoder is used in the siamese network, we freeze the trained encoder layers, and we add another trainable layer. This way we ensure the use of the trained representation while fine-tuning the model to minimize the BPR loss.


Conv2D, input:(224,224,3), output:(222,222,16)
MaxPooling2D, input:(222,222,16), output:(111,111,16)
Conv2D, input:(111,111,16), output:(109,109,32)
MaxPooling2D, input:(109,109,32), output:(54,54,32)
Conv2D, input:(54,54,32), output:(52,52,32)
MaxPooling2D, input:(52,52,32), output:(26,26,32)
Conv2D, input:(26,26,32), output:(24,24,64)
MaxPooling2D, input:(24,24,64), output:(12,12,64)
Conv2D, input:(12,12,64), output:(10,10,64)
MaxPooling2D, input:(10,10,64), output:(5,5,64)
Conv2D, input:(5,5,64), output:(1600)
MaxPooling2D, input:(1600), output:(147)
Reshape, input:(147), output:(7,7,3)
Conv2D_Transpose, input:(7,7,3), output:(14,14,64)
BatchNormalization, input:(14,14,64), output:(14,14,64)
Conv2D_Transpose input:(14,14,64), output:(28,28,64)
BatchNormalization, input:(28,28,64), output:(28,28,64)
Conv2D_Transpose, input:(28,28,64), output:(56,56,32)
BatchNormalization, input:(56,56,32), output:(56,56,32)
Conv2D_Transpose, input:(56,56,32), output:(112,112,32)
BatchNormalization, input:(112,112,32), output:(112,112,32)
Conv2D_Transpose, input:(112,112,32), output:(224,224,16)
Conv2D, input:(224,224,16), output:(224,224,3)

(b) decoder architecture

(a) encoder architecture

Figure 5.1: Convolutional Autoencoder Architecture

The complete process is shown in figure 5.2. On the left side, the convolutional autoencoder training process is presented. After it's trained, we take only the encoder, freeze its layers, add a trainable layer (in the size of K as in [5]) and use it in the siamese network as presented on the right side of the figure.

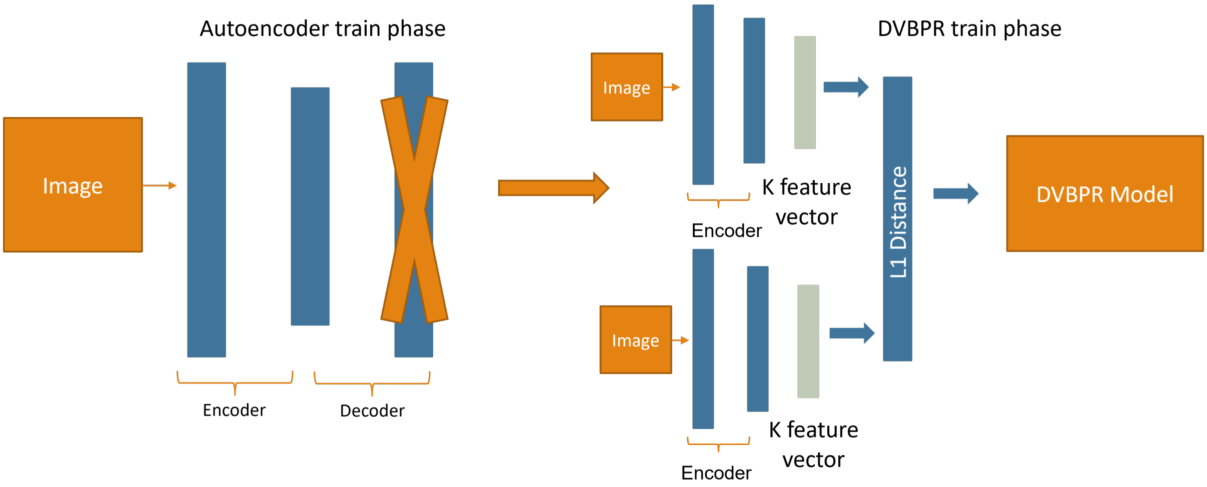


Figure 5.2: Convolutional Autoencoder solution training process

5.2 ImageNet Based Solution

Our second solution consists of using a model that was trained on the ImageNet dataset. We used the InceptionV3 model suggested by [12] that was proved to get very good results in the image classification task. Our inspiration to use this approach is that we hypothesize that this way we can represent the items' images by their category. By doing so, we expect to get recommendations to the user based on the items' categories they interacted with, e.g. if a user loves shoes, we will recommend them items from the same category.

In this approach, we use the InceptionV3 pre-trained model when removing it's last softmax layer- this way we get a vector that represents the image. Next, we conduct a transfer learning approach by freezing all the InceptionV3 model layers and adding one trainable layer in the size of K. This way the siamese network learns to minimize the BPR loss but only fine-tunes the ImageNet model, which ensures the use of the learned weights on the greater data of the ImageNet dataset.

The complete process of this approach is shown in figure 5.3. On the left side, the removal of the last softmax layer is presented. We then freeze all of the ImageNet model's layers and we add a trainable layer in the size of K as shown on the right side of the figure on each siamese leg.

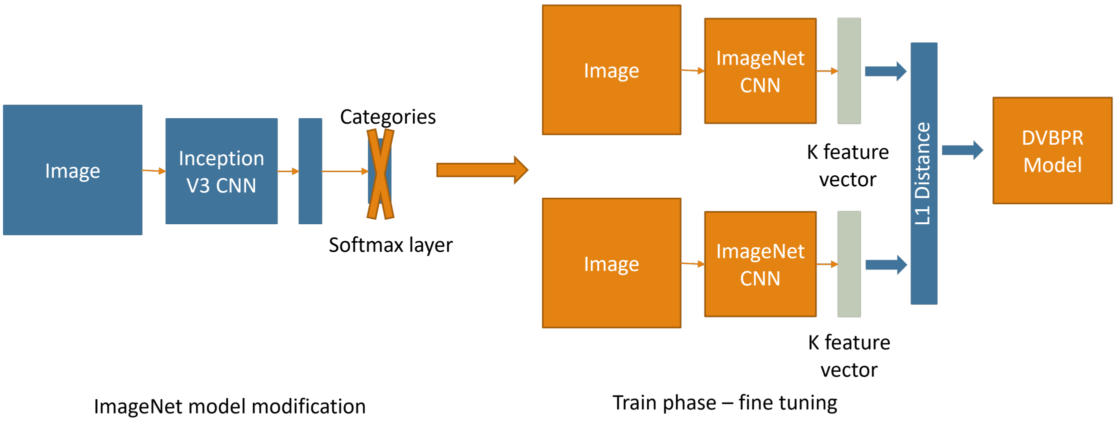


Figure 5.3: ImageNet based solution training process

6 Evaluation

6.1 Datasets

In our experiments, we use three datasets that were used in the DVBPR paper [5] - AmazonFashion and AmazonMen which consist of reviews of clothing items crawled from Amazon.com [5] and the third one was crawled from Tradesy.com, which is a c2c online platform for buying and selling used fashion items [5]. For all datasets, we used an implicit feedback (review or click were treated as positive feedback). In addition, each item on each dataset has exactly one associated image.

The datasets were split into train, validation and test sets. Each user, on the train set, has at least 3 items that he interacted with in order to eliminate cold items' problem, and the validation and test set contain exactly one item that the user was interacted with. We always report performance for the model that achieved the best performance on our test set. Statistics of our datasets are given in Table 6.1.

Dataset	#Users	#Items	#Interactions
Amazon Fashion	45184	166270	358003
Amazon Men	34244	110636	254870
Tradesy.com	33864	326393	723137

Table 6.1: Datasets statistics

6.2 Experimental Plan

At our training phase, we first sampled for each user an item that he was interacted with, and another item that he was not. We extracted the images for each item and inserted them into our Siamese network together with the user ID. For efficiency, we split the data into batches, while the batch size selected from {64,128,256}. We found out that the performance of all batch sizes were similar, but the running time was different, so we chose a batch size of 128 samples which gave the fastest results.

In addition, at our validation and test phases, we used the same batch size, but the sampling was little different - because these sets contain only one item that the user was interacted with, we chose this one item as observed one, and we sample from the non-observed items another one. We perform optimization by stochastic gradient ascent using the Adam optimizer with learning rate of 0.0001. We trained the model for 20 to 25 epochs, while each epoch ran approximately 10 minutes, so the total training time was ≈ 3.5 hours. Moreover, for the CAE-based model, we trained an autoencoder separately for 3 epochs.

The model hyperparameters - regularization and latent dimensions, were tuned by grid search and based on the hyperparameters that were used in [5]. The best hyperparameters we found were: $\lambda_{\Theta_u} = 1$ - the regularization of the user vector, $\lambda_{\Theta} = 0.001$ - the regularization of the model weights, $latentDim = 50$. All experiments of our method were conducted on Google Colab using GPU accelerator.

6.3 Evaluation Metrics

We calculate the AUC to measure recommendation performance of our method and that of baselines. The AUC measures the quality of a ranking based on pairwise comparisons (and is the measure that BPR-like methods are trained to optimize). Formally [5], we have

$$AUC = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|D_u|} \sum_{(i,j) \in D_u} \varsigma(x_{u,i} > x_{u,j})$$

where $D_u = \{(i,j) | (u,i) \in Test_u \wedge (u,j) \notin (Observed_u)\}$ and $\varsigma(\cdot)$ is an indicator function. In other words, we are counting the fraction of times that the 'observed' items i are preferred over 'non-observed' items j.

7 Results

To compare our methods to the DVBPR method we conducted different experiments on the same datasets and we report the performance in terms of the AUC metric as done in the compared paper of Kang et al. [5]. The reported results are of the performance on the test set of each dataset and can be seen in table 7.1. We can see that overall the CAE-based model has relatively poor results compared to the other methods, however the InceptionV3-based model has relatively close results to the original DVBPR model. In each of the datasets the best model score is boldfaced.

Dataset	DVBPR	CAE-Based	InceptionV3-Based
Amazon Fashion	**0.783**	0.52	0.764
Amazon Men	**0.74**	0.51	0.723
Tradesy.com	**0.78**	0.508	0.721

Table 7.1: InceptionV3-Based model and CAE-Based model recommendation performance compared to DVBPR model in terms of the AUC (higher is better). The best performing method in each row is boldfaced.

As mentioned before, the models were trained on Google Colab environment, which uses limited hardware and also has limited usage time. In our opinion one can get much better results with our methods by using them on better environments that can cope running for a longer time with greater number of epochs. This way in our opinion, the InceptionV3-based model could even get better than the DVBPR model.

8 Discussion

The results of the CAE-based model were relatively poor, but the results of the InceptionV3-based model were relatively close to the DVBPR ones. The limited hardware we could run our experiments on were a big limitation in our opinion on the experimental results. One could get better results with better and stronger hardware.

We also think that the architecture of the Inception-V3 model that was proved to be very good for classification tasks and was trained on huge data, can get better representations of the images than the encoder of the CAE that we built. In the future work we think that creating better architectures for the CAE can boost its results and get comparable results to the other methods.

Different architectures could be applied and might have gotten better results too. In our method we add one trainable layer when more could be added in different ways. We also use the second last layer from the InceptionV3 model when other layers could be utilized. Thus, an interesting future work can be in the architectures of the trainable layers, the architecture of the CAE and the utilized layer from the InceptionV3 model (or even a different pre-trained model).

9 Conclusion

In this study, we tried to improve methods of visually-aware recommender systems that use implicit feedback such as in the work of Kang et al. [5]. We tried different solutions- a CAE-based solution and an ImageNet-based one. The second mentioned solution has managed to get very close results to the DVBPR [5] study and in our opinion could even get better with better hardware and different architectures. Thus, in the future work we would suggest to try different architectures of the ImageNet fine-tuning part, new architectures of the CAE model and even try to incorporate different ImageNet based models.

Bibliography

[1] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[2] W. Gong and L. Khalid. Aesthetics, personalization and recommendation: A survey on deep learning in fashion. *arXiv preprint arXiv:2101.08301*, 2021.

[3] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[4] S. Hiriyannaiah, G. Siddesh, and K. Srinivasa. Deep visual ensemble similarity (dvesm) approach for visually aware recommendation and search in smart community. *Journal of King Saud University-Computer and Information Sciences*, 2020.

[5] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley. Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 207–216. IEEE, 2017.

[6] G. Koch, R. Zemel, R. Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[7] M. Maggipinto, C. Masiero, A. Beghi, and G. A. Susto. A convolutional autoencoder approach for feature extraction in virtual metrology. *Procedia Manufacturing*, 17:126–133, 2018.

[8] F. J. Martinez-Murcia, A. Ortiz, J.-M. Gorriz, J. Ramirez, and D. Castillo-Barnes. Studying the manifold structure of alzheimer's disease: A deep learning approach using convolutional autoencoders. *IEEE Journal of Biomedical and Health Informatics*, 24(1):17–26, 2019.

[9] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[10] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.