Article

Predicting Implicit User Preferences with Multimodal Feature Fusion for Similar User Recommendation in Social Media

Jenq-Haur Wang ^{1,*} , Yen-Tsang Wu ¹ and Long Wang ^{2,*}¹ Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 106344, Taiwan; buddyswu@gmail.com² School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

* Correspondence: jhwang@csie.ntut.edu.tw (J.-H.W.); lwang@ustb.edu.cn (L.W.)

Abstract: In social networks, users can easily share information and express their opinions. Given the huge amount of data posted by many users, it is difficult to search for relevant information. In addition to individual posts, it would be useful if we can recommend groups of people with similar interests. Past studies on user preference learning focused on single-modal features such as review contents or demographic information of users. However, such information is usually not easy to obtain in most social media without explicit user feedback. In this paper, we propose a multimodal feature fusion approach to implicit user preference prediction which combines text and image features from user posts for recommending similar users in social media. First, we use the convolutional neural network (CNN) and TextCNN models to extract image and text features, respectively. Then, these features are combined using early and late fusion methods as a representation of user preferences. Lastly, a list of users with the most similar preferences are recommended. The experimental results on real-world Instagram data show that the best performance can be achieved when we apply late fusion of individual classification results for images and texts, with the best average top-k accuracy of 0.491. This validates the effectiveness of utilizing deep learning methods for fusing multimodal features to represent social user preferences. Further investigation is needed to verify the performance in different types of social media.

Keywords: deep learning; user preference learning; feature fusion; similar user recommendation

Citation: Wang, J.-H.; Wu, Y.-T.; Wang, L. Predicting Implicit User Preferences with Multimodal Feature Fusion for Similar User Recommendation in Social Media. *Appl. Sci.* **2021**, *11*, 1064. <https://doi.org/10.3390/app11031064>

Received: 23 December 2020

Accepted: 21 January 2021

Published: 25 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In popular online social networks such as Twitter, Facebook, and Instagram, it is easy for users to share information and post their opinions and comments. Given the huge amount of user-generated content (UGC), it is difficult to search for the most relevant information effectively. Since people tend to share information that interests them and comment on the topics they like, user posts and comments are likely to reflect their preferences. In addition to individual posts, it would be useful if we can also recommend groups of people with similar interests. This might help users discover more relevant content in social media. Conventionally, characteristics of users are often defined by their demographic information, such as gender, age group, occupation, and income status. These are usually obtained by filling in surveys and basic personal information which is often lacking since users are reluctant to provide this in most social media. Thus, it is a challenge to learn detailed user preferences without explicit user feedback.

Conventional recommender systems focus on suggestion of items as a function of similar items or users in previous records. However, user behaviors in online social networks are different from e-commerce websites since there is no explicit feedback from users regarding the relevance assessment of items. Instead, users participate in social media by posting, replying (or retweeting), and forwarding multimedia content. Since users might post various types of multimedia content related to the corresponding texts

in the surrounding paragraphs, it would be useful if these contextual relations could be utilized to discover implicit user preferences. In this paper, we propose a multimodal feature fusion approach to implicit user preference prediction by learning deep features from texts and images in user posts for recommending similar users in social media. First, text and image features in user posts are extracted using deep learning techniques, such as convolutional neural networks (CNNs) and TextCNN. Then, these features are combined using multimodal feature fusion methods as a potential representation of user preferences. Lastly, similarity among user preferences can be defined and calculated for finding and recommending similar users. Since there are multiple types of features, in this paper, we compare the effects of early and late fusions of features for user preference prediction. To evaluate the performance of user recommendation, a dimensionality reduction method using autoencoders is compared with user clustering.

From the experimental results of Instagram data, we can see the clear advantage of deep learning and feature fusion over individual features. Recommending the top- k similar users when applying late fusion of individual classification results of texts and images gives the best average top- k accuracy of 0.491. This shows the potential of discovering implicit user preferences from multimodal content posted by users. Further investigation is needed to verify the effects in different types of social media.

The major contributions of this paper can be summarized as follows:

- First, we propose a convolutional deep learning method for extracting image and text features from user posts as a potential representation of implicit user preferences.
- Second, we compare feature fusion methods to combine text and image features from user posts for predicting user preferences.
- In our experimental results on real-world Instagram data, the best average top- k accuracy of 0.491 for recommending top-50 similar users can be obtained when applying late fusion on text and image features. This shows an improvement of 36.3% over the baseline in terms of accuracy.

2. Related Works

There are two categories of studies related to our work. One involves techniques that model user preferences by items and user's information for user recommendation. The other involves deep learning techniques adopted for feature extraction and fusion.

For user or people recommendation in social networks, most existing approaches rely on social relations and network structures in addition to content similarity. For example, Chen et al. [1] found out that the social network structure tends to give known contacts, while content similarity helps to find new friends. Hannon et al. [2] considered content-based techniques and collaborative filtering approaches based on followees and followers of users.

Armentano et al. [3] proposed to recommend relevant users by exploring the topology of the network. Since content-based approaches tend to have low precision while collaborative filtering based approaches based on follower-followee relations have data sparsity issues, Zhao et al. [4] proposed a community-based approach that utilizes an Latent Dirichlet Allocation (LDA)-based method on follower-followee relations to discover communities before applying matrix factorization for user recommendation. Gurini et al. [5] proposed to extract semantic attitudes from user-generated content, including sentiment, volume, and objectivity, and they conducted people recommendation using matrix factorization. In this paper, since there are no social network structures available, we used content-based approach as our baseline model. Moreover, we included word embedding using a pretrained Word2Vec model to get semantic information of texts.

Predicting user preferences is very important when constructing a recommender system. In social networks, user preferences can be derived from three sources. The first is user post contents including texts and images. They provide the direct evidence of what users like. The second is user interested topics, which could be reflected from tags in posts. The third is the user relations in social networks. Recent preference prediction models

integrate information including user posts [6], images [7,8], social network attributes [9], and user demographic information [10]. Examples include gender, age, and political tendency [11]. In past research, researchers considered integrating information such as reviews and social relations to predict user preferences. For example, some methods learned user sentiments from user reviews [12,13] and item topics regarding user preferences [14,15]. Some methods used hybrid methods to learn user's opinion in different domains [8,16]. Unlike the above studies, we did not use the extra information such as social relations or user demographics. Instead, we focused on extracting information from post contents, including texts and images.

On the other hand, deep learning practices have been applied to texts to provide more insights into the reasons behind users' preferences and more awareness of item features they consider relevant [17]. Deep learning has been shown to be effective in user preference learning [18,19]. Palangi et al. [20] proposed a deep model which used a variant of recurrent neural network (RNN) architecture called Long Short-Term Memory (LSTM) for retrieval task. The study by Tai [21] used Tree-LSTM to predict semantic connection of sentences and sentiment classification. Yousif et al. [22] combined CNN and Bidirectional LSTM (Bi-LSTM) to analyze citation sentiment and purpose classification. Seo et al. [23] used CNN with attention to model user preferences and item properties as expressed in review texts. However, these methods take more computation due to their complex structure. In contrast, our proposed method focuses on fusing different features instead of relying on complex models to achieve good performance in classification.

Some existing preference prediction methods utilized users' review texts to learn user preference. In Chambua et al. [24], they used the hybrid approach to learn and represent user preferences and predict them by using RNN-LSTM and probabilistic matrix factorization in the Amazon Products Datasets. Because they only used review texts as the user feature, they are faced with the problem of missing data. In Lv et al. [25], the visual and social features were fused by linear regression, matrix factorization, and support vector regression. Unlike our proposed method, they used tags and titles as the textual feature. This might not accurately capture the user's true emotional preferences. Zhang et al. [26] used the attention mechanisms to extract textual and visual features by a variant of CNN called VGGnet and LSTM in the Flickr dataset. Through a linear attention mechanism, they fused the textual, visual, and user features for prediction. With the same problems as Lv et al. [25], the textual description of posts only included titles and tags which might not reflect emotional preferences. Aloufi et al. [27] used the visual and social features and information-associated content to predict popular images by ranking the Support Vector Machine (SVM) model in the Flickr dataset. They added several extra features to predict which photos would be popular. Mazloom et al. [28] used the Instagram dataset for their experiment. They used features from users, items, and contexts of posts as representation to predict the popularity of a post related to a specific user and item in social media by matrix factorization. The difference to our proposed method is that they added the visual and textual sentiment based on Visual Sentiment Ontology [29] and SentiStrength [30]. Unlike the above papers, we represent user preferences by fusing text contents and image features extracted with convolutional neural networks. Furthermore, we address the issue of user recommendation by autoencoders and user clustering.

Feature fusion is an important method in pattern recognition which allows for more robust predictions by incorporating multiple features that might complement each other. When some of the features are missing, we can still make predictions. Contextual similarity [31] has been extensively exploited recently in retrieval tasks, such as biological information retrieval, natural image search, shape retrieval, and analysis of time series. A more recent example is the unsupervised ranking model in which all words of a query or document are embedded into vectors, which are matched by deep neural networks [32]. The multimodal approach was constructed by different input sources [33,34]. For example, the multimodal approach based on image and text features was employed in multiple tasks such as retrieval, classification, and natural language processing. To find relation-

ships between text and image features, two general multimodal fusion approaches were deployed: early and late fusions [35]. Features with poor performance will greatly affect the effects of early fusion [36]. In contrast to early fusion, late fusion uses mechanisms such as averaging [37], voting [38], and learned model [39,40] to fuse predictions from each model. In our paper, we aimed to compare the effects of early and late fusions in user preference prediction.

3. The Proposed Method

In this paper, we propose a feature fusion approach to implicit user preference learning from user posts and related images for similar user recommendation. The overall system architecture is illustrated in Figure 1.

Figure 1. The system architecture of our proposed method.

As shown in Figure 1, there are three modules in the proposed method: feature extraction and embedding, feature fusion, and user recommendation. First, user posts in social networks are collected as the dataset, where texts and images are extracted by using text and image convolutional networks, respectively. Second, text and image features are combined using feature fusion techniques as the representation of user preferences. Lastly, on the basis of the similarity among user preferences, top similar users are recommended by autoencoders and user clustering.

3.1. Feature Extraction

People often express their opinions on selected topics by posting related texts and images they are interested in. These behaviors might show their implicit user preferences. Thus, to better understand what people like, in this paper, we assume that user preferences can be represented by the characteristics of texts and images in their posts and comments.

Previous studies used the title or tags as text features for classification. However, user preferences cannot be effectively learned since tags might be ambiguous and diverse in their meanings. Since text contents usually contain more semantic information such as emotions or stances than tags, text contents are used instead of tags. In this paper, we focus on deep learning methods such as the TextCNN model [41] for text feature extraction. First, text documents are represented by a word embedding model such as Word2Vec [42,43]. This is a distributional representation of words among different contexts in fix-sized vectors. Next, we use TextCNN [41] to extract text features. This is a slight variant of the CNN architecture by Collobert et al. [44]. The TextCNN architecture is shown in Figure 2.

Figure 2. Convolutional neural networks (CNNs) for text classification [41].

As shown in Figure 2, there are four layers in the TextCNN model: word embedding layer, convolutional layer, max pooling layer, and fully connected layer. First, given a sentence of length n , we denote it by its word embeddings as follows:

$$x_{1:n} = x_1 \oslash x_2 \oslash \dots \oslash x_n, \quad (1)$$

where x_i is the embedding of word i with dimension k , and $\oslash$ is the concatenation operator. Second, in the convolutional layer, the convolution operation involves a filter $w \in R^{h \times k}$, which is applied to a window of h words to produce a feature c_i for the i -th window as follows:

$$c_i = f(w * x_{i:i+h-1} + b), \quad (2)$$

where $b \in R$ is a bias term and f is a nonlinear function, such as the hyperbolic tangent. For the given sentence $x_{1:n}$, this filter produces a feature map c as follows:

$$c = [c_1, c_2, \dots, c_{n-h+1}], \quad (3)$$

where $c \in R^{n-h+1}$. Then, in the max pooling layer, we take the maximum value $c' = \max(c_1, c_2, \dots, c_{n-h+1})$ as the feature map corresponding to this particular filter. Lastly, the fully connected layer outputs the final classification result or category. In this paper, we also extract outputs from the layer before the last layer to compare the effects of feature fusion.

To extract image features, we utilize a CNN model called VGG16 [45] to convolve the image pixels represented in the Red-Green-Blue (RGB) color model. This architecture contains several differences from the previous convolutional networks. It is a multilayer convolutional network that is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows a significant improvement by increasing the depth to 16 layers (13 convolutional layers and three fully connected layers). Layers of configurations are designed using the same principles, inspired by Ciresan et al. [46] and Krizhevsky et al. [47]. Compared with other models such as [48], VGG16 is a CNN architecture with fewer parameters but comparable accuracy. For this reason, it is used as the Image CNN model to extract the feature map from images.

3.2. Feature Fusion

With texts and images in a user post, we try to combine them in two different ways, i.e., early fusion and late fusion.

3.2.1. Early Fusion

In early fusion, embeddings of texts and images are input to the same CNN model simultaneously for feature extraction. Early fusion is also known as feature-level fusion, which can be expressed as follows: $x_m = f(x_1, \dots, x_n)$, where an aggregated representation x_m of features is computed by function f that integrates individual features $x_1, \dots, x_n$. Early fusion combines different input sources into a single feature vector, which is used as inputs to the classification framework. The advantage of early fusion is that it learns all the features in one phase. This makes the training pipeline easier, but a lot of important information might be lost. The architecture is shown in Figure 3.

Figure 3. The architecture for early fusion of text and image features.

As shown in Figure 3, first, since the sizes of text and image embeddings are different, we use a fully connected (FC) neural network to fuse these vectors. There are two different ways of concatenation: text vector concatenated with images or image vector concatenated with texts. Next, we extract the output of different layers in the VGG16 model including the feature map and the final classification result or category as the combined representation for the texts and images.

3.2.2. Late Fusion

In late fusion, text and image features are extracted and then combined for classification. This is also known as decision-level fusion [49]. It integrates different model predictors by a fusion mechanism to come up with the final decision. Late fusion can be expressed as follows: $output = g(f_1(x_1), \dots, f_n(x_n))$, where functions $f_1, \dots, f_n$ are applied to individual features and function g is used to aggregate the individual decisions by $f_1, \dots, f_n$. The main disadvantage of late fusion is that it cannot learn the correlation among features. Compared to early fusion, late fusion tends to be more robust to features that have negative influence [36]. The architecture is shown in Figure 4.

Figure 4. The architecture for late fusion of text and image features.

As shown in Figure 4, after embedding texts and images into vectors by Word2Vec and RGB models, respectively, they are input to two separate CNN models for feature extraction as described in Section 3.1. The corresponding feature map and text/image category can be obtained. Then, to fuse text and image features, in addition to using a fully connected layer, we also try to combine the two feature maps by inner product since they are of the same size. The idea is to increase the correlation and reduce the dimension through the inner product.

3.3. User Preference Learning

After combining text and image features using either early or late fusion techniques, we obtain the corresponding feature map and category for each user post. To further represent user preferences, there are two different methods. First, since feature map is the internal representation of a post, the centroid of the feature maps of all posts can be regarded as the user preferences. We can simply add all features with the addition operator to get the centroid. Second, the category of each user post is assumed to reflect part of the user preferences since people tend to post topics they are interested in. People with the same preferences are more likely to post in the same category. Therefore, for each user, we count the number of posts in each category and find out the majority category as the user preference. These are shown in Equations (4) and (5).

$$f_c = F_a([c_1, c_2, \dots, c_n]), \quad (4)$$

$$f_R = F_c([R_1, R_2, \dots, R_n]), \quad (5)$$

where F_a is the addition operator and F_c is the count operator, whereas c_i and R_i are the feature maps and categories of the post, respectively. Since the output categories R_i in neural networks are usually represented by one-hot encoding, we can simply accumulate them using the count operator to find out the majority category. By aggregating the feature

map and category of all images and texts posted by the user, we can obtain the potential feature for user preferences.

3.4. Dimension Reduction

From preliminary observations, we found that feature maps in images and texts usually have very high dimensions, which could be as high as 8192 after feature fusion. In addition to computational time complexity, it may also include a lot of noise inside. To find out the most important features, we further apply autoencoders for dimension reduction.

Given a fixed dimensional representation of the user preferences as inputs, we design the autoencoder as a single-hidden-layer d -dimensional neural network. Autoencoders consist of an encoder that transforms input to a code, and a decoder which reconstructs the input from the code. When the number of neurons in the hidden layer of autoencoders is less than that in the input layer, it is forced to learn the compressed representation of the input data. After dimensionality reduction, our model performs similarity calculations among users to generate a list of recommendations. The architecture is shown in Figure 5.

Figure 5. The user preference learning architecture with dimension reduction.

Since the goal of using autoencoders is to reduce dimensionality by simple neural networks, it would be beneficial if the dimensionality can be reduced while maintaining comparable performance in recommendation. In order to find out the most compact representation which is still effective in recommendation, we conducted experiments to verify the effects of different dimensionality d on the performance of user recommendation.

3.5. Recommendation Based on User Clustering

When there are many users, it takes a lot of time to calculate similarity between each one according to the aforementioned method. It is hard to give real-time recommendation. Thus, we further investigate the effect of clustering all users by the K-means algorithm to reduce the number of candidates for similar user recommendation.

Although K-means clustering is incremental, to save time when updating clusters, we propose to divide it into two stages: offline clustering and online recommendation. First, users are represented by the output category after learning user preferences in Section 3.3 and then clustered using the K-means algorithm. Since the dimension of the category vector is lower than that of feature maps, it reduces the computational time. Then, we calculate cosine similarity by using the feature map of each member in the same cluster to find out the most similar members to recommend. Because the number of users in each group is much smaller than all users, it also reduces the computational time for online recommendation if user clustering can be done offline.

If the original user posts a new post, we only need to calculate the similarity for the members of the same group. If we have a new member, we just need to calculate which cluster it belongs to for recommendation and then calculate the similarity between cluster members, instead of all users. In addition, since clustering can be done offline, the method greatly improves the efficiency of online recommendation. The architecture is shown in Figure 6.

Figure 6. The architecture of recommendation based on user clustering.

The problem for conventional K-means algorithm is to determine the best number of clusters. Since Cascade K-means [50,51] can automatically determine the optimal number of clusters, it was selected to solve this problem. We use the classification result of the user as inputs for user clustering with Cascade K-means. Lastly, after calculating cosine similarity among the members in the same cluster, we can obtain the most similar K members to recommend.

Although we did not deploy it as a potential real-world recommendation system, we could implement different types of thresholds before initiating the update process by clustering. For example, we could count the number of new users, and set the threshold for a new user ratio. Furthermore, when the percentage of new posts by a user reaches a threshold, we could initiate an update process to update the cluster members for more accurate recommendation.

4. Experiments

In this section, we describe our datasets, how training and testing are performed, and our baseline algorithms, and we analyze the experimental results.

To evaluate the performance of the proposed method, we used the current mainstream social media—Instagram—in our experiments. To the best of our knowledge, there is no publicly available social media dataset that includes both images and texts posted by the same user. To collect the Instagram dataset, we randomly selected six hashtags from the list of top 100 hashtags on Instagram in 2018 as queries to get user posts. This can be done using the Hashtag Search API, which is only available for Instagram Professional accounts. Thus, we had to directly crawl the latest updates for each user on Instagram webpages for tag-based queries. There is a limit of a maximum number of 45 posts per user that can be crawled. To avoid large variations in user participation where there were too few posts to learn the features, we chose to keep only users who had posted at least nine posts, which is a 5:1 ratio (20% of the maximum). Each post contained the user identifier (ID) and contents of the user posts and hashtags. This dataset was used to train the classification models of user preferences (denoted as Dataset_1). There were 239 users and 6941 posts in Dataset_1, from which we split into 5553 posts for training and 1388 posts for testing. The major language was English.

Considering the diversity in linguistic form, we used two pretrained Word2Vec embeddings as dictionaries for English and Chinese. The detailed attributes of the word embeddings are outlined in Table 1.

Table 1. The detailed attributes of the Word2Vec embeddings.

Word Embedding	Language	Property	Dimensionality
GoogleNews-vectors	English	Universal language	300
PTT2015-2018	Chinese	Verbal language	300

4.1. The Performance of User Preference Learning

First, we compared the performance of three classification models: TextCNN, ImageCNN, and early fusion of the two. The training data in Dataset_1 were used to build the models, and the test data in Dataset_1 were used to evaluate the models. In the dataset, there were only user-defined hashtags in addition to texts and images in user posts. It is difficult to know user preferences without explicit user feedback. To obtain the ground truth for user preferences, the hashtag that was used to retrieve each post was set as the class label of that post. We assumed that users implicitly express their preferences through the use of hashtags in the posts and comments that they are interested in.

In our ImageCNN model, the training procedure generally followed Simonyan et al. [45]. In this model, we included three convolutional layers with the kernel size set to 5 and strides = 1. The training was regularized by weight decay (the L2 penalty multiplier was set to 5×10^{-4}) and dropout regularization for the first two fully connected layers with a dropout rate of 0.2. The learning rate was initially set to 0.1. Rectified Linear Unit (ReLU) and Root Mean Square Error (RMSE) were used as the activation function and loss function. The final layer is the softmax layer. The performance of the ImageCNN model is shown in Figure 7.

Figure 7. The accuracy of the ImageCNN model.

As shown in Figure 7, when the dimension was 4096, we could obtain the best performance with an accuracy of 98.89%. Since the pretrained model of VGG16 had dimensionality in multiples of 1024, models larger than 4096 could not be run due to the limitation of our computer hardware. In the subsequent experiments, we used the dimension of 4096 for the ImageCNN model.

In our TextCNN model, the training procedure generally followed Kim [41]. We used Word2vec for embedding text contents as input. Three filters of window sizes 3, 4, and 5 were tested, the stride was 1, the dropout rate was set to 0.5, the L2 constraint was 3, and the mini-batch size was set to 50. The learning rate was set to 0.1. We applied a max pooling operation. Training was done through stochastic gradient descent over shuffled mini-batches with the Adadelta update rule [41]. RMSE and LeakyReLU were used as the loss and activation function, respectively. We chose an adaptive learning rate method for neural networks proposed by Geoff Hinton called RMSPROP as the optimizer. The performance of the TextCNN model is shown in Figure 8.

As shown in Figure 8, when the dimension was 2100, we could obtain the best performing TextCNN model with an accuracy of 96.8%. In the subsequent experiments, we used the dimension of 2100 for the TextCNN model. Then, when texts and images were input at the same time, we applied early fusion as shown in Figure 3. The best performance with an accuracy of 80% could be observed when the dimension was 4096. According to our observations, the lower accuracy might have been due to the number of classes in multiclass classification.

Figure 8. The accuracy of the TextCNN Model.

4.2. The Performance of Similar User Recommendation

To evaluate the performance of user recommendation and to deal with posts in different languages, we further collected a larger dataset (denoted as Dataset_2) where 12 tags were randomly selected from the list of top 100 hashtags on Instagram in 2018 as queries. There were 1143 users and 49,353 posts in Dataset_2, where the major language was Chinese.

In order to compare the performance of different models in similar user recommendation, we established the ground truth for similar users as follows: in the absence of real user preferences, we assumed that the hashtags in each user's posts represents the user's implicit preference and attributes. Thus, the set of hashtags $Tags(u_i)$ for each user u_i was collected from all the posts as the real preference. The similarity between users was calculated by Jaccard similarity as follows:

$$\text{sim}(u_i, u_j) = \text{Jaccard}(Tags(u_i), Tags(u_j)). \quad (6)$$

Then, the ground truth of top k similar users for a given user u_i was defined as follows:

$$SL_k(u_i) = \{u_j | \text{argmax}_k \text{sim}(u_i, u_j)\}, \quad (7)$$

where $j \neq i$. That is, top k users with the highest similarity to the given user were regarded as the ground truth.

With the recommendation ground truth, we then defined the evaluation metric of the top- k accuracy of a user u_i by calculating the Jaccard similarity between the recommendation list $RL_k(u_i)$ generated by the proposed method and the ground truth $SL_k(u_i)$, as follows:

$$\text{Accuracy}_k(u_i) = \frac{RL_k(u_i) \cap SL_k(u_i)}{RL_k(u_i) \cup SL_k(u_i)}. \quad (8)$$

Then, the overall performance of the model was calculated by the average of the top- k accuracies for all users. To compare the performance of different models, we evaluated the average top- k accuracies for each model when recommending top k similar users, for different values of k .

4.2.1. The Baseline Models

In this paper, we took a different approach to recommender systems than conventional ones. Conventional recommender systems have the premise that the ratings of different users on various items are available. Then, standard recommendation methods such as user-based and item-based collaborative filtering approaches can be used to learn similar users or similar items on the basis of their past rating behaviors, so that ratings for items that were never given before can be deduced. However, in a real-world scenario in social networks where only post content and hashtags are available, it is not possible to obtain the "items" since posts are only posted by one user and replied or shared by a few. It

is not possible for them to be “rated” by many different users. In this case, we propose a more realistic approach using hashtags as user preferences for recommending similar users in social networks. When we only have post content and hashtags, the baseline for such a recommendation system is simply content-based recommendation from posts since hashtags have been used as the ground truth.

In our experiments, we considered two baseline models: `baseline_text` and `baseline_pic`. The `baseline_text` model is based on the representation of user preferences only by the text contents of users’ posts. Specifically, for each user, we simply use the distributed representation of the texts posted by that user as their characteristics. First, we converted the words into vectors using the same Word2Vec embedding model with the same dictionaries in Table 1. Next, vectors for all words in a document were averaged as the representation of the document. We further averaged the vectors of all documents posted by the user as the user feature. Lastly, we calculated the cosine similarity of the current user to all other users to generate a list of recommendations. On the other hand, the `baseline_pic` model is based on the representation of user preferences only by the image contents of user posts using the ImageCNN model. The only difference is that we took the featuremaps and categories from all images posted by the user as the user feature.

4.2.2. Effects of the CNN Models

In order to verify the performance of CNN models for extracting text and image features, we compared the average top-k accuracy for the TextCNN model and ImageCNN model with the `baseline_text` model. First, when using word embedding from Word2Vec for texts, the performance of the `baseline_text` model on Dataset_1 achieved the average top-k accuracy of 0.128. For the TextCNN model, the average top-k accuracy was 0.326 and 0.322 when the number of dimensions was 2100 and 4096, respectively. For the ImageCNN model, the average top-k accuracy was 0.330 and 0.327 when the number of dimensions was 2100 and 4096, respectively. When we combined the Word2Vec embedded text features and ImageCNN extracted image features by a simple concatenation, the best performance of the `baseline_pic` model could be obtained with the average top-k accuracy of 0.331. This showed a slightly better performance when using multimodal features than that of individual features. Both TextCNN and ImageCNN models achieved better performance than the `baseline_text` model. This validates the effectiveness of using CNN models to extract text and image features for finding similar users.

4.2.3. Effects of Multimodal Feature Fusion

To validate the effectiveness of multimodal feature fusion, we further compared the performance of combining text and image features with CNN models using early fusion and late fusion. In early fusion, we concatenate text and image features as inputs to a single CNN model, while, in late fusion, we separately input text features and image features into TextCNN and ImageCNN models, with the resulting feature map and categories are combined as the user feature for recommendation. The performance comparison is shown in Figure 9.

As shown in Figure 9, the best performance of early fusion could be obtained when the number of dimensions was 2100, with the average top-k accuracy of 0.473. For late fusion, the best performance could be obtained with the average top-k accuracy of 0.491, when we used the categories as features, with the number of dimensions of 4096. We could observe the better performance of both early fusion and late fusion than the `baseline_pic` model with a simple concatenation. This shows the effectiveness of feature fusion, especially for late fusion. In the case of late fusion, we further address the effect of k on the average top-k accuracy and number of hits in Figure 10.

Figure 9. The performance comparison of early and late fusions.

Figure 10. The average top-k accuracy and number of hits for different values of k.

As shown in Figure 10, the average number of hits in our recommendation increased when k increased. Moreover, we could obtain the best average top-k accuracy of 0.491 when k = 50. This was also the best performance in all of our experiments. In the remaining experiments, we used the value of k as 50.

4.2.4. Effects of Autoencoder for Dimension Reduction

To verify the effectiveness of using an autoencoder in finding similar users, we took the features produced by either early fusion or late fusion as inputs to a single-layer neural network for dimension reduction. The performance comparison of autoencoders in different dimensions is shown in Figure 11.

Figure 11. The performance of autoencoders with different dimensionality.

From the experimental results in Figure 11, the best performance could be obtained with the average top-k accuracy of 0.462, when the feature dimension was reduced to 1500 if we used the feature map as the features. This was not as good as the best performance after either early or late fusion. However, the number of dimensions could be greatly reduced from 4096 to 1500. This can help reduce the storage needed for the redundant features. It would be useful to keep the more important features for more efficient computations in practice.

4.2.5. Effects of User Clustering

To verify if user clustering is effective in finding similar users, we further took the feature map and categories from either early fusion or late fusion. Since user clustering can be done offline, only the similarity calculations are needed for those users in the same cluster as the given user. From our experimental results, the best number of clusters was 3 with the average top-k accuracy of 0.367. Since the effectiveness was not as good as a direct recommendation, we further verify the efficiency of recommendation using user clustering in Table 2.

Table 2. The comparison of late fusion and user clustering in terms of execution time. N/A, not applicable.

Model	Number of Recommended Users	Time for Feature Extraction (ms)	Time for Recommendation (ms)	Time for Clustering (ms)	Total Time (ms)
Late fusion	238	2,943,698	507,654	(N/A)	3,451,352
Clustering	238	2,943,698	177,318	1,342,126	4,463,124

As shown in Table 2, although the total amount of time needed in the user clustering model was longer than late fusion due to the time for clustering, the actual amount of time for recommendation was less than that for late fusion. In practical application scenarios, user clustering only needs to be calculated offline once when there are certain amounts of new content or new users. This makes it much faster in the recommendation stage.

To investigate the effects of our proposed approach on a larger scale, we further conducted experiments on Dataset_2. The performance comparison of different models on both datasets is summarized below.

As shown in Table 3, multimodal feature fusion is useful since we can combine clues from both text and image contents. The best performance could be obtained for the late fusion model, with the average top-k accuracy of 0.491 and 0.281 for Dataset_1 and Dataset_2, respectively. Although the accuracy could not be further improved by autoencoders, we could obtain comparable performance with the benefits of dimensionality reduction since a lower dimension means less storage needed, as well as less time required for recommendation systems.

Table 3. The performance comparison of all experimental results.

Model	Dimension	Performance of Dataset_1	Performance of Dataset_2
Baseline_text	300	0.128	0.098
Baseline_pic	2100	0.331	0.130
TextCNN	2100	0.326	0.172
ImageCNN	4096	0.327	0.173
Early fusion	2100	0.473	0.223
Late fusion	4096	0.491	0.281
Autoencoder	1500	0.462	0.245
User clustering	4096	0.367	0.207

5. Discussions

From these experimental results, there are several observations. First, the performance of the CNN-based models (either text or image) is better than the baseline model (text). The improvement of TextCNN or ImageCNN over the baseline is 19% in terms of accuracy. This

shows that neural network-based models can improve the performance of content-based baseline models.

However, in this paper, we focused on the task of user recommendation. Existing recommender systems focus on item recommendation on the basis of user ratings on different items. User recommendation is more challenging since we have many users, where each one has different attributes and preferences that cannot be exactly the same as others. When we categorize user preferences, multiclass classification is more difficult than binary classification. To the best of our knowledge, there is no publicly available dataset for user recommendation. Since there is no ground truth in real-world data, we assume that hashtags posted by users represent their implicit user preferences. For each user, the users with more similar hashtags are assumed to be the ground truth of their similar users. Then, we evaluate the performance by the average top-k accuracy of each system-generated user list, as shown in Equation (8). As shown in Section 4, all four proposed model variations outperformed the baseline model. Among the four different variations, late fusion achieved the best performance of 0.491 in terms of average top-k accuracy for similar user recommendation. This shows the effectiveness of the proposed model. Although autoencoders cannot further improve the accuracy, comparable performance can be obtained with the benefits of dimensionality reduction. This gives better efficiency for real-world recommendation systems.

To further assess the performance on a larger scale, all models were executed on Dataset_2. We applied the same parameters as obtained in the best performing models in previous experiments for Dataset_2. From the experimental results, the best performance could be obtained with the average top-k accuracy of 0.281 for late fusion when we used the categories as the user feature with a dimension of 4096. It is worth noting that the proposed approach is far better than the baseline. The improvement across all models is about 8–18%. This further validates the usefulness of our proposed method in practice. The attributes in Dataset_2 are completely different from those in Dataset_1 since they were written in different languages. This shows the effectiveness of our proposed approach in different languages.

The reasons why the performance of Dataset 2 was inferior to that of Dataset 1 are as follows: first, since more tags were used as our queries to Instagram to collect our Dataset 2, we could obtain more posts by more people which could give us more diverse content. Specifically, we included hashtags in Chinese, which involves cross-lingual issues when we cannot effectively identify different hashtags in different languages with similar meanings. One of the reasons why the top-k accuracy for user recommendation was not as high as accuracy in classification tasks is that user-provided hashtags are very diverse. Since our approach is evaluated on the basis of hashtags, this shows its limitation. However, as shown in the experimental results, our proposed approach can outperform the cases of single-modal feature. We demonstrated the effectiveness of multimodal feature fusion from texts and images for user recommendation. In the future, we plan to resolve the issue of diverse hashtags by consolidating the semantic meanings of hashtags using word embedding models or state-of-the-art deep learning models for linguistic tasks such as transformers or Bidirectional Encoder Representations from Transformers (BERT).

As a potential application of our proposed approach, we can utilize the similar user recommendation algorithm to build an intellectual data crawler in social networks. For example, according to the targeted topics of interest, we can utilize our proposed approach to discover related posts, with multimedia content and related user information. Then, by clustering similar users on the basis of user preferences, it would be useful to further expand our proposed approach across multiple social networks that might be different in their structure. This could help reduce the problem of social network analysis across different social networks.

In real application scenarios, there could be issues such as posted and reposted images, as well as drawings and photos of the same thing, to name a few. In this paper, we did not distinguish between posted and reposted images if they were captured in the same

resolution. Since the features of all images were extracted by the same CNN architecture of VGG16, the posted and reposted images would have the same features if they are represented by the same pixels. This classifies them into the same category, from which our recommendation is made. Following the same line of thought, various content such as drawings and photos of the same thing would not have exactly the same features since it would be difficult to mimic the photos when drawing the same thing. However, CNN models were demonstrated [52] to improve the performance of sketch-based image retrieval (SBIR) by extracting deep features in recent years. Since we utilize VGG16 with multiple convolutional and pooling layers, our algorithm is able to extract the important semantic features from drawings and photos of the same thing that give similar classification results.

6. Conclusions

In this paper, we proposed a multimodal feature fusion approach to user preference learning by combining user post contents and related image features for recommending similar users on a popular social platform, Instagram. With the help of convolutional networks, the features of images and texts can be effectively extracted. Using either early or late fusion methods, these features can be effectively integrated as a representation of user preference. According to the experimental results, our proposed method achieved good performance on the real-world datasets. Firstly, the effectiveness of our scheme was better than the conventional method of distributed representation of texts only. Secondly, the late fusion approach combining images and texts obtained the best performance of 0.491 in terms of average top-k accuracy. Thirdly, the proposed method could also be applied in datasets with different language attributes. This shows the effectiveness and potential of our proposed approach to similar user recommendation in social networks. Further investigation is needed to verify the performance of our proposed approach on different social media platforms.

Author Contributions: Conceptualization, J.-H.W. and L.W.; methodology, Y.-T.W.; software, Y.-T.W.; writing—original draft preparation, Y.-T.W.; writing—review and editing, J.-H.W. and L.W. All authors read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the research grants of the Ministry of Science and Technology, Taiwan under the grant number MOST109-2221-E-027-090 and partially funded by the National Taipei University of Technology–University of Science and Technology Beijing Joint Research Program under the grant number of NTUT-USTB-107-07. It was also partially funded by the National Applied Research Laboratories, Taiwan under the grant number of NARL- ISIM-109-002.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy concerns as stated in Instagram Data Policy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, J.; Geyer, W.; Dugan, C.; Muller, M.; Guy, I. “Make New Friends, but Keep the Old”—Recommending People on Social Networking Sites. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI’09), Boston, MA, USA, 4–9 April 2009; pp. 201–210.
2. Hannon, J.; Bennett, M.; Smyth, B. Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches. In Proceedings of the fourth ACM Conference on Recommender Systems (RecSys’10), Barcelona, Spain, 26–30 September 2010; pp. 199–206.
3. Armentano, M.G.; Godoy, D.; Amandi, A. Topology-based recommendation of users in micro-blogging communities. *J. Comput. Sci. Technol.* **2012**, *27*, 624–634. [[CrossRef](#)]
4. Zhao, G.; Lee, M.L.; Hsu, W.; Chen, W.; Hu, H. Community-Based User Recommendation in Uni-Directional Social Networks. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management (CIKM’13), San Francisco, CA, USA, 27 October–1 November 2013; pp. 189–198.

5. Gurini, D.F.; Gasparetti, F.; Micarelli, A.; Sansonetti, G. Temporal people-to-people recommendation on social networks with sentiment-based matrix factorization. *Future Gener. Comput. Syst.* **2018**, *78*, 430–439. [[CrossRef](#)]
6. Lee, R.K.-W.; Hoang, T.-A.; Lim, E.-P. On analyzing user topic-specific platform preferences across multiple social media sites. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1351–1359.
7. Li, X.; Pham, T.-A.N.; Cong, G.; Yuan, Q.; Li, X.-L. Krishnaswamy S Where you instagram? Associating your instagram photos with points of interest. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; pp. 1231–1240.
8. Caicedo, J.C.; Kapoor, A.; Kang, S.B. Collaborative personalization of image enhancement. In Proceedings of the CVPR 2011, Providence, RI, USA, 20–25 June 2011; pp. 249–256.
9. Wei, H.; Zhang, F.; Yuan, N.J.; Cao, C.; Fu, H.; Xie, X.; Rui, Y.; Ma, W.-Y. Beyond the words: Predicting user personality from heterogeneous information. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 305–314.
10. Long, Y.; Lu, Q.; Xiao, Y.; Li, M.; Huang, C.-R. Domain-specific user preference prediction based on multiple user activities. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 3913–3921.
11. Volkova, S.; Coppersmith, G.; Van Durme, B. Inferring user political preferences from streaming communications. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; Volume 1, pp. 186–196.
12. Yousif, A.; Niu, Z.; Tarus, J.K.; Ahmad, A. A survey on sentiment analysis of scientific citations. *Artif. Intell. Rev.* **2019**, *52*, 1805–1838. [[CrossRef](#)]
13. Nilashi, M.; Ahani, A.; Esfahani, M.D.; Yadegaridehkordi, E.; Samad, S.; Ibrahim, O.; Sharef, N.M.; Akbari, E. Preference learning for eco-friendly hotels recommendation: A multi-criteria collaborative filtering approach. *J. Clean. Prod.* **2019**, *215*, 767–783. [[CrossRef](#)]
14. Tarus, J.K.; Niu, Z.; Kalui, D. A hybrid recommender system for e-learning based on context awareness and sequential pattern mining. *Soft Comput.* **2018**, *22*, 2449–2461. [[CrossRef](#)]
15. Wan, S.; Niu, Z. An e-learning recommendation approach based on the self-organization of learning resource. *Knowl. Based Syst.* **2018**, *160*, 71–87. [[CrossRef](#)]
16. Wan, S.; Niu, Z. A Hybrid E-learning Recommendation Approach Based on Learners' Influence Propagation. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 827–840. [[CrossRef](#)]
17. Zhang, Y.; Lai, G.; Zhang, M.; Zhang, Y.; Liu, Y.; Ma, S. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval, Gold Coast, QLD, Australia, 6–11 July 2014; pp. 83–92.
18. Barkan, O. Bayesian neural word embedding. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 3135–3143.
19. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.-S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
20. Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; Ward, R. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 694–707. [[CrossRef](#)]
21. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015.
22. Yousif, A.; Niu, Z.; Chambua, J.; Khan, Z.Y. Multi-task learning model based on recurrent convolutional neural networks for citation sentiment and purpose classification. *Neurocomputing* **2019**, *335*, 195–205. [[CrossRef](#)]
23. Seo, S.; Huang, J.; Yang, H.; Liu, Y. Representation learning of users and items for review rating prediction using attention-based convolutional neural networks. In Proceedings of the SIAM Conference on Data Mining 2017, Houston, Texas, USA, 27–29 April 2017.
24. Chambua, J.; Niu, Z.; Zhu, Y. User preferences prediction approach based on embedded deep summaries. *Expert Syst. Appl.* **2019**, *132*, 87–98. [[CrossRef](#)]
25. Lv, J.; Liu, W.; Zhang, M.; Gong, H.; Wu, B.; Ma, H. Multi-feature fusion for predicting social media popularity. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 1883–1888.
26. Zhang, W.; Wang, W.; Wang, J.; Zha, H. User-guided hierarchical attention network for multi-modal social image popularity prediction. In Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee, Lyon, France, 23–27 April 2018; pp. 1277–1286.
27. Aloufi, S.; Zhu, S.; El Saddik, A. On the prediction of flickr image popularity by analyzing heterogeneous social sensory data. *Sensors* **2017**, *17*, 631. [[CrossRef](#)] [[PubMed](#)]
28. Mazloom, M.; Hendriks, B.; Worringer, M. Multimodal context-aware recommender for post popularity prediction in social media. In Proceedings of the Thematic Workshops of ACM Multimedia 2017, Mountain View, CA, USA, 23–27 October 2017; pp. 236–244.
29. Borth, D.; Ji, R.; Chen, T.; Breuel, T.; Chang, S.-F. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In Proceedings of the 21st ACM International Conference on Multimedia 2013, Barcelona, Spain, 21–25 October 2013; pp. 223–232.

30. Thelwall, M.; Buckley, K.; Paltoglou, G.; Cai, D.; Kappas, A. Sentiment strength detection in short informal text. *J. Am. Soc. Inf. Sci. Technol.* **2010**, *61*, 2544–2558. [[CrossRef](#)]
31. Donoser, M.; Bischof, H. Diffusion processes for retrieval revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1320–1327.
32. Dehghani, M.; Zamani, H.; Severyn, A.; Kamps, J.; Croft, W.B. Neural ranking models with weak supervision. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 65–74.
33. Gallo, I.; Calefati, A.; Nawaz, S. Multimodal classification fusion in real-world scenarios. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 36–41.
34. Kiela, D.; Grave, E.; Joulin, A.; Mikolov, T. Efficient large-scale multi-modal classification. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 5198–5204.
35. Baltrušaitis, T.; Ahuja, C.; Morency, L.-P. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 423–443. [[CrossRef](#)] [[PubMed](#)]
36. Lan, Z.-Z.; Bao, L.; Yu, S.-I.; Liu, W.; Hauptmann, A.G. Multimedia classification and event detection using double fusion. *Multimedia Tools Appl.* **2013**, *71*, 333–347. [[CrossRef](#)]
37. Shutova, E.; Kiela, D.; Maillard, J. Black holes and white rabbits: Metaphor identification with visual features. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 160–170.
38. Morvant, E.; Habrard, A.; Ayache, S. Majority vote of diverse classifiers for late fusion. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 153–162.
39. Lv, P.; Meng, X.; Zhang, Y. FeRe: Exploiting influence of multi-dimensional features resided in news domain for recommendation. *Inf. Process. Manag.* **2017**, *53*, 1215–1241. [[CrossRef](#)]
40. Khatibi, A.; Belém, F.M.; Da Silva, A.P.C.; Almeida, J.; Gonçalves, M. Fine-grained tourism prediction: Impact of social and environmental features. *Inf. Process. Manag.* **2020**, *57*, 102057. [[CrossRef](#)]
41. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:14085882.
42. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, arXiv:1310.4546.
43. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:13013781.
44. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
45. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:14091556.
46. Ciresan, D.C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, 16–22 July 2011; pp. 1237–1242.
47. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, 1097–1105.
48. Qassim, H.; Verma, A.; Feinzimer, D. Compressed residual-VGG16 CNN model for big data places image recognition. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 169–175.
49. Atrey, P.K.; Hossain, M.A.; El Saddik, A.; Kankanhalli, M.S. Multimodal fusion for multimedia analysis: A survey. *Multimedia Syst.* **2010**, *16*, 345–379. [[CrossRef](#)]
50. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. Theory Methods* **1974**, *3*, 1–27. [[CrossRef](#)]
51. Hamerly, G.; Elkan, C. Alternatives to the k-means algorithm that find better clusterings. In Proceedings of the Eleventh International Conference on Information and Knowledge Management, McLean, VA, USA, 4–9 November 2002; pp. 600–607.
52. Yu, D.; Liu, Y.; Pang, Y.; Li, Z.; Li, H. A multi-layer deep fusion convolutional neural network for sketch based image retrieval. *Neurocomputing* **2018**, *296*, 23–32. [[CrossRef](#)]