Towards a Privacy Respecting Image-based User Profiling Component

Panagiotis Galopoulos, CERTH-ITI, Thessaloniki, Greece, gpan@iti.gr
Chryssanthi Iakovidou, CERTH-ITI, Thessaloniki, Greece, c.iakovidou@iti.gr
Vasiliki Gkatziaki, CERTH-ITI, Thessaloniki, Greece, vasgat@iti.gr
Symeon Papadopoulos, CERTH-ITI, Thessaloniki, Greece, papadop@iti.gr
Yiannis Kompatsiaris, CERTH-ITI, Thessaloniki, Greece, ikom@iti.gr

Abstract—This paper explores the development of a framework for content-aware user profiling, studying how image producers and consumers can be better understood and consequently better served through services such as matchmaking and friend recommendations. User interests and similarities are extracted and analyzed on the edge employing state of the art CNN models over user images for the tasks of classification, as well as, building latent user representations from personal media content. A private-by-design approach is employed through the development and deployment of models on-device, avoiding the need for communicating personal data to a central service. Experimental results show that user profiling can provide accurate ranking of the users’ interests and meaningful user associations through profile similarity.

Index Terms—user profiling, user matching, image annotation, privacy-by-design, MobileNet

I. INTRODUCTION

The challenge of effectively understanding user interests and traits through their online behaviour becomes ever more apparent as users spend a growing amount of time being online, e.g., consuming multimedia content, searching for information related to a given interest or otherwise socially interacting. With mobile internet usage being at an all-time high, information and content stored on a user’s device can ultimately contribute to building very effective and precise representations of the user, their preferences and behavioural patterns. Understanding the particularities of a user can significantly boost the appeal and effectiveness of applications, allowing the optimisation and personalisation of their service. For instance, in the context of the HELIOS\(^1\) H2020 project, we are building a user profiling framework to provide content aware matching between users. User profiling is, however, a multi-faceted problem and the issues one must address include the choice of information that is to be utilized, the decision of whether to use existing standards or to create new ones while also accounting for privacy issues related to storing and processing personal data and information.

Mobile user profiling can be roughly categorized as follows: i) explicit profile extraction, where users are profiled utilising explicitly defined mobile data e.g., demographics, website clicks, mobile purchases, in-app behaviours [1], and ii) implicit profile learning, where methods employ collaborative filtering, latent factor models, network embedding and deep learning to build a user representation [2]. The effectiveness of methods that fall into the explicit profile extraction category depends highly on the accurate collection of comprehensive user-related information and thus often suffers in terms of consistency and scalability. On the other hand, implicit profiling often suffers from information scarcity (especially true for methods that depend on abundant peers’ information to group users [3] or demand domain knowledge to regularize the employed models and avoid overfitting (e.g. matrix/tensor factorization methods that model profiles as latent factors and learn user profiles through optimization within a large parameter space [4]). More recent works have applied deep neural networks to learn the network embeddings of users for many end-to-end deep learning tasks, such as deep learning based recommender systems [5] [6].

In this work, we focus on utilizing image content present in the user’s device (e.g. stored, liked, exchanged images) for building user profiles. For this to work effectively we need to be able to capture the semantics of the images instead of simply what is depicted, and to this end we present two distinct methods. The first depends on using explicitly defined concepts that are deemed relevant to the user profiling task and training a hierarchical classifier to recognize them. The second method, on the other hand, attempts to learn an appropriate user representation from user similarity data that we infer from the autotag expansion pack of the YFCC100m dataset [7]. Using Deep Metric Learning (DML) our model learns a user embedding that relates the distance of the embeddings to the similarity of the users.

While appropriately defining the nature of user profiles is essential, there is also a need for proper datasets to train new deep learning models. The challenge arises due to lack of publicly available, large, high quality datasets of user images annotated with relevant information to accommodate the task of user profiling. Thus, we build two datasets for the training and evaluation purposes of this work. The first one, described in section IV-A0a and [8], contains images from Pinterest categorized in predefined interest classes as well as 12 randomly selected Pinterest users, whose pinboards have been manually labeled to fit the aforementioned interest classes, for testing purposes. The second one, described in section IV-A0b, is based on a subset of the YFCC100m dataset

\(^1\)https://helios-h2020.eu/
We also want to safeguard the privacy of the user data and thus require that the proposed models can run on the users' mobile devices without communication to external servers. This design decision limits the memory and computation resources available to our models and requires a right trade-off between performance and efficiency. To summarize, the contributions of this work include the following:

- the development of a hierarchical classification scheme for user profiling;
- a novel formulation of a DML model to calculate pairwise user similarities;
- the construction of a publicly available dataset to be used for user profiling or similar tasks;
- a study on the feasibility of developing such models in a privacy respecting manner on mobile devices.

II. RELATED WORK

A. Profiling with Predefined Categories

Using predefined categories to construct user profiles is the most straightforward way to proceed, offering the benefits of simplicity and interpretability. A common approach is based on detecting common objects and recognizing scenes in users’ image collections, as is done in [9], [10].

Compared to object and scene categories, interest categories, are more directly related to the users and thus constitute a promising alternative. In our work, we opted for this route and the specifics of our interest-based model will be fleshed out in section III-A. The approach followed in [11], is similar to the classification method we present in that they trained a model based on interest categories with data scraped from Pinterest. However, they did not make their dataset publicly available and the technique they used to collect the data depended on the pinboards being annotated to the appropriate category, a label that is no longer available from Pinterest. The categories were also not fine-tuned to better represent the task, the classification was flat without hierarchical elements and there was no effort to make the models appropriate for mobile use.

B. Profiling with Mined Representations

While profiling with predefined categories has the benefits of being more straightforward and interpretable, the extracted information is inevitably limited by the fixed nature of the categories. To address this limitation, methods that learn the appropriate user representation from data have been proposed [12]–[14]. In particular, DML techniques have been employed in [14] to train a user similarity network, as we also do in section III-B, but using a different formulation.

In the remainder of this section we will provide a brief introduction to DML. Metric Learning is the field that focuses on learning a distance function to measure the similarity between data samples. Recent works combine Metric with Deep Learning for learning a distance function through the training of a Deep Neural Network (DNN). The main objective of such works is to approximate an embedding function that maps samples in a feature space where relevant samples are closer to each other than to irrelevant ones. A DNN can approximate such an embedding function through a training scheme that penalizes the violation of the samples’ ordering.

Any deep learning architecture can be selected and adapted based on the underlying problem for the implementation of the DNN. Several DML setups have been proposed in the literature for the training of the DNN [15]–[18].

Considerable effort has also been invested over the years into the critical step of the DML process that is the organization of the data samples in the form required by the loss function and the composition of a representative training set. We will follow the semi-hard negative mining scheme [16], where along with the hard negatives [19], all those negative samples are also considered that are further away from the anchor than a fixed positive sample, but still closer than the anchor-positive distance plus the margin, offering thus a softer transition between positive and negative samples and significantly boosting the overall performance.

III. METHODOLOGY

This section presents the proposed framework for content-based user profiling. First we describe the profiling through the extraction of user interests from predefined categories introducing the concept of interest categories. Next we focus on building a method able to capture more semantic nuances turning away from predefined concepts and instead exploiting user similarity data to learn a latent user representation.

A. Profiling based on Predefined User Interests

Our design strategy starts by trying to classify the users’ images in interest categories and interpret the interest distribution of each user’s images as their profile. To achieve this we first need to define the interest categories. Since there are not any suitable publicly available datasets, we created one from scratch, utilizing the online platform Pinterest, as it is a popular social network where users post content that reflects their interests.

Inspired from the Pinterest topics (now rebranded as ideas2) we selected the 15 interest categories shown in Table I and also further split four of the most popular ones into subcategories, displayed in Table II.

Table I

| Categories | Images | Categories | Images |
|------------|--------|------------|--------|
| Fashion    | 8279   | Tattoos    | 3424   |
| Entertainment | 7897 | Health-Fitness | 3238 |
| Art        | 6362   | Architecture | 1876   |
| Sports     | 5263   | Technology | 1783   |
| Animals    | 4649   | Travel     | 1685   |
| Vehicles   | 4454   | Plants-Flowers | 1653 |
| Food-Drinks| 4370   | Kids-Babies | 1641   |
| Home decor | 3892   | Total      | 61,993 |

2https://www.pinterest.com/ideas/
TABLE II
Compound Interest Categories

| Fashion   | Entertainment   | Sports       | Vehicles  |
|-----------|----------------|--------------|-----------|
| Jewellery | General        | Basketball   | Planes    |
| Nails     | Music          | Soccer       | Boats     |
| Makeup    | Books          | Gymnastics   | Trucks    |
| Bags      | Video Games    | Tennis       | Motorcycles|
| Hair      | Movie-Series   | Swimming     | Cars      |
| Clothes   |                 | Football     |           |
| Shoes     |                 |              |           |

We define two levels of detail in order to provide better flexibility and accuracy, given that there is a natural hierarchy in the interest categories and the images of a compound category tend to share a lot of similarities making it often difficult to accurately identify the subcategories. For example, clothes and bags are two subcategories of fashion, but an image of a woman holding a bag can be misleading regarding what constitutes the implied interest. Furthermore, the flexibility of being able to compute both coarse and fine profiles is important in a mobile setting where memory and computation resources are valuable. This hierarchical scheme is implemented by first training a coarse classifier on the 15 categories of Table I and then training another four local classifiers according to Table II.

A thresholding step is also introduced to account for the fact that not all the available images of a user are expected to be exploitable for profiling. It is quite possible that some will not convey any useful information about the interests of a user, the most obvious example being photos that were taken by accident, as well as, blurry and distorted photos. To increase the system’s robustness against such noisy pictures we make use of a filtering mechanism that rejects the images that our model classifies as not informative enough. A non-informative image in our case would correspond to an image that the model cannot clearly classify to some interest(s). From an information theory perspective, the uninformative images would be those that maximize the entropy [20] of the output probability distribution, which occurs when the latter approximates the uniform distribution. In such cases, the output probability distribution does not present a clear peak and the model does not confidently predict a category. Thus, when the probability distribution is found to be entirely below a predefined threshold, the model should disregard the image.

Fig. 1 and 2 show the inference process for a coarse and fine profile, respectively. Let us assume that a user has \( N \) images \( I_i, i = 1, \ldots, N \). To calculate the user’s coarse profile each image is passed to the coarse classifier, \( c_{\text{coarse}} \), to produce the per image coarse category distributions \( d_{\text{coarse}}^i \).

\[
d_{\text{coarse}}^i = c_{\text{coarse}}(I_i), i = 1, \ldots, N
\]  

Next, the filtering procedure is activated to disregard images with distribution below the threshold \( t \) and the final coarse user profile is calculated as the average of the distributions of the remaining images. That is, let \( \mathcal{I}_t \) be defined as

\[
\mathcal{I}_t = \{ i \in \{ 1, \ldots, N \} \mid (\exists j) d_{\text{coarse}}^i[j] \geq t \}
\]  

then the user’s coarse profile at threshold \( t \) is

\[
p_{\text{coarse}}^t = \frac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} d_{\text{coarse}}^i
\]  

To calculate the fine distribution of the image \( I_i \), that is \( d_{\text{fine}}^i \), we can simply replace the index that each compound category appears in \( d_{\text{coarse}}^i \) with the output of the corresponding local classifier multiplied by the coarse probability of the compound category. As an optimization to avoid running all four local classifiers for every input image, we can run the local classifier only if the the coarse value of the corresponding compound category is above the defined threshold \( t \). In all other cases, we can assume that the output of the local classifier is a non-informative uniform distribution. The final fine profile, \( p_{\text{fine}} \), is then calculated as the average fine distribution over all images, similarly to the case of coarse categories.

**B. Profiling through User Similarity and DML**

While the proposed hierarchical classification scheme described in section III-A has the attractive property of building profiles that correspond to meaningful human concepts related to predefined interest categories, it also carries the shortcomings of limited semantic variability and scalability. This is to be expected as it is often the case that model expressiveness is at odds with model interpretability. In this section we propose a method that is in principle able to capture significantly more semantic nuances. To do that we employ user similarity data to construct a latent user representation that retains as much of the original similarity structure as possible. Learning from
similarity data closely resonates with the field of DML and as such we propose the use of the triplet loss during training. Triplet loss requires batches of triplets that consist of an anchor user (reference user) along with a similar and a dissimilar user. This way the model learns to build user representations that reflect the original user similarity structure and can effectively be interpreted for our purposes as user profiles. Because the user representations are not trained on manually specified concepts, the model has the capacity to discover through the available data the most suitable way to represent the users.

To evaluate this method we created a dataset, described in section IV-A0b, based on YFCC100m [7] and its autotags expansion, that includes for each user a collection of images and a 1570-d vector that represents the distribution of autotags in their images. We then define the similarity $S_{ij}$ between users $i$ and $j$ to be the cosine similarity of their autotag distribution vectors. Because these vectors correspond to probability distributions they have unit length and are positive and as such the cosine similarity becomes the inner product of the two vectors, for which it holds that $0 \leq S_{ij} \leq 1$ and $S_{ij} = S_{ji}$.

DML is focused on learning a distance function to measure in our case the similarity between users by approximating an embedding function that maps users in a feature space where similar users are close, while dissimilar ones are further apart.

Let us assume that we have a user $x$, that we will refer to as the anchor user, and two other users $x^+$ and $x^-$ that are similar and dissimilar to the anchor, respectively. Then we will call the pair $(x, x^+)$ a positive sample, the pair $(x, x^-)$ a negative sample and the triplet $(x, x^+, x^-)$ will be the input to our model during training. To train our model we have to choose a strategy to mine these triplets as well as an appropriate embedding to a user space where similar users are close and dissimilar ones are further apart.

An appropriate loss function would take high values when $x^-$ is closer to $x$ than $x^+$ is and low values when the opposite happens, with a reasonable margin separating the two. A formulation that reflects this is

$$
\mathcal{L}_{tr} = \max(0, \mathcal{D}(x, x^+) - \mathcal{D}(x, x^-) + m)
$$

where $\mathcal{D}$ is the distance function we approximate with our model and $m > 0$ is a margin parameter to ensure a sufficiently large difference between the anchor-positive and negative distances. This is known as the triplet loss.

Having settled on the triplet loss function, we consider an appropriate sampling strategy to generate the triplets at each iteration of the training process. We note that training on all the possible $O(n^3)$ triplets would be infeasible. The ideal triplets are those that do not trivially satisfy the loss constraint, but rather violate it and thus provide valuable feedback to the training process. We specifically chose to use semi-hard triplets, which are the triplets $(x, x^+, x^-)$ that produce

$$
0 < \mathcal{L}_{tr}(x, x^+, x^-) < m
$$

That is, samples that satisfy $\mathcal{D}(x, x^+) < \mathcal{D}(x, x^-)$ as desirable, but not within the appropriate margin $m$ (chosen equal to 1 in our implementation). The triplets are created online and so it is also important to ensure that each batch has enough semi-hard examples. For this reason, we use a large batch size of 512 and create the triplets by first selecting all the $(x, x^+)$ pairs within the batch and then for each such anchor-positive pair we select a negative sample $x^-$ such that the semi-hard rule is satisfied. Our labels, however, only assign a similarity score between 0 and 1 for each user pair and thus a threshold needs to be defined to translate these scores to positive and negative examples. For our implementation user pairs with a similarity score above 0.8 are marked as positive and those with similarity below 0.4 as negative.

Last, we specify the architecture of the model, shown in Fig. 3, that will approximate the distance function $\mathcal{D}$ between the users. The primary input to the model is each user’s images, from which we extract features with a pretrained CNN and minimize the computational overhead by calculating the mean of the produced vectors. This first part of the network is not specifically trained for the task due to the lack of available large scale data. The trainable part is the fully connected network that follows and consists of three linear layers with ReLU activations in between. The output of this network is the user embeddings, which are used as inference as the user profiles, and the distance between the embeddings is defined by computing their Euclidean distance.

IV. Evaluation Setup

A. Datasets

First we briefly discuss the datasets that we created for training and testing our models; we have also made PID2020 (IV-A0a) publicly available following an anonymization process [8].

a) Pinterest Interest Dataset 2020 (PID2020): This dataset was used for training and testing the hierarchical classification method of section III-A. To build the training part, for each of the categories shown at Tables I and II we queried Pinterest with several relevant terms. For the testing part we had to resort to manual labeling of user profiles. So, for 12 Pinterest users we manually classified their Pinboards in one of the defined categories or if they did not correspond to any of the categories we left them unlabeled. The user
profile was calculated by assigning to each image the category of the Pinboard it belonged to and constructing the category distribution. We note that while unlabeled images were left out during the construction of the ground truth profile, they were included in the test set as noise that we consider realistic to be present in real cases; it is the model’s job to filter them out with the thresholding mechanism previously described.

b) YFCC100m Autotag Expansion: While user tags vary widely and not always offer any meaningful semantic content, machine tags are predictable and easier to reason with. For this reason we utilized the autotag expansion of the YFCC100m to define user profiles as the autotag distribution of the user’s images. Approximately 170,000 users are included and a total of 5 million images, taken from the MediaEval2016 Benchmark [21], a subset of the YFCC100m dataset. This dataset was used for training the DML model described in section III-B.

B. Backbone Network

The backbone network is the CNN responsible for the majority of the memory and computation load. Because we are interested in deploying the proposed models to mobile devices, we chose to experiment with MobileNetV2 [22]. We also compare its performance with EfficientNet-B3 [23], a state-of-the-art network. Both networks are pre-trained on ImageNet [24] and fine-tuned end-to-end for the hierarchical classification task of section III-A. The DML model, described in section III-B, is not trained end-to-end, but rather only the features of the images are extracted from the backbone CNN and we then train the fully connected layer that we defined on top of the features. The described architecture is deliberately one of the simplest possible, but at the same time efficient, in order to be able to run in a wide range of mobile devices. To port the trained networks to mobile devices, we convert them to TFLite [25] objects and also experiment with quantizing their weights.

V. Experiment Results

TABLE III

| Network      | Coarse | Fashion | Entert. | Sports | Vehicles |
|--------------|--------|---------|---------|--------|----------|
| EfficientNet-B3 | 70.7   | 92.8    | 67.7    | 81.6   | 90.3     |
| MobileNetV2   | 66.7   | 91.7    | 64.7    | 76.2   | 87.3     |
| Quantized TFLite | 62.8   | 86.8    | 54.0    | 64.5   | 87.0     |

Table III shows the top-1 accuracy of the coarse and the local classifiers as measured on the left-out validation set after a 0.8/0.2 training/validation split of the PID dataset (IV-A0a). As expected, the best performing model is the one with the EfficientNet-B3 backbone, but MobileNetV2 follows closely. While, the quantized TFLite model lags pretty significantly behind in some categories, the overall classification accuracy seems to be reasonably close.

However, classification only matters for our purposes as a means to construct user profiles and as such we test our classifiers at producing coarse and fine profiles for 12 manually labeled users. Since we are mainly interested in the ranking ability of our models, that is whether they are able to pick out a user’s order of preference and not so much at the absolute values, we chose evaluation metrics typically used for ranking tasks. We report the Area Under Curve (AUC), the Mean Average Precision (MAP) and the Normalized Discounted Cumulative Gain (NDCG) for the coarse and fine profiles at Tables IV and V, respectively.

TABLE IV

| Network           | AUC Mean | AUC Std | MAP Mean | MAP Std | NDCG Mean | NDCG Std |
|-------------------|----------|---------|----------|---------|-----------|----------|
| EfficientNet-B3   | 93.3     | 4.15    | 87.6     | 8.76    | 93.0      | 7.00     |
| MobileNetV2       | 94.4     | 4.82    | 89.8     | 6.57    | 94.3      | 5.25     |
| Quantized TFLite  | 93.6     | 4.86    | 88.5     | 7.42    | 92.5      | 6.30     |

TABLE V

| Network           | AUC Mean | AUC Std | MAP Mean | MAP Std | NDCG Mean | NDCG Std |
|-------------------|----------|---------|----------|---------|-----------|----------|
| EfficientNet-B3   | 95.4     | 5.13    | 87.0     | 11.10   | 92.8      | 6.87     |
| MobileNetV2       | 95.0     | 6.73    | 87.3     | 11.40   | 91.6      | 7.66     |
| Quantized TFLite  | 91.4     | 5.94    | 74.3     | 15.92   | 82.9      | 16.79    |

The results seem promising; however, the sample size of 12 users is very small, which is reflected on the high standard deviations. Further investigation is needed, but finding appropriate data for this task is very challenging.

Assessing the performance of the DML model is also not straightforward, but we devised an evaluation scheme as follows. From the left-out validation set we collected the 1000 most common user tags and created a bag-of-words representation for each user. This time we did not use the machine generated tags, but rather the user defined tags that accompanied the images, in order to eliminate the effects of the autotagging process from our results. Based on these tags we calculate the Jaccard similarity between two users as the ratio of the size of the intersection to the size of the union of the tags of a user pair. We are interested in measuring the Jaccard similarity, between each user and the k-th most similar user according to our model, calculated using the Euclidean distance between the respective user embeddings. We expect a decreasing trend for the Jaccard similarity as k increases, since the users become more dissimilar, and this is also reflected in their tags. Our hypothesis is indeed validated in Fig. 4.

![Fig. 4. Evaluation of the DML model. We observe a correlation between the predicted similarity and the Jaccard similarity of the users.](image-url)
VI. CONCLUSIONS AND FUTURE WORK

This paper explored the problem of developing content aware user profiling in a mobile setting. Our first approach relied on predefined interest categories, leading to the design of a deep learning model capable of inferring both coarse and fine grained user profiles based on users’ photo collections. The model was trained with a newly collected dataset based on image and user data achieving scores of 93.6, 88.5 and 92.5 mean AUC, MAP and NDCCG respectively using a small, efficient TFLite model suitable for mobile deployment. Furthermore, the content aware user profiles created in this way are interpretable as they correspond to meaningful concepts, an important feature when aiming to be transparent with the users about how the underlying algorithms work.

To address the disadvantage of relying only on predefined categories, we created an additional model with lower interpretability, but with the potential to capture more concepts with higher semantic coverage. This model was based on DML and approximates a function that can map a user’s photo collection to an embedding space where similar users are closer and dissimilar users are further apart. The model was trained on a subset of the YFCC100m dataset annotated with autotags, while it was also demonstrated that there is a correlation between the closeness of the users’ embeddings and the similarity of the users based on the tags they provided to their photos.

It is important to note that care has been taken to design models that are capable of being deployed in a mobile environment, that is they can be compressed in a small binary package and require modest computational resources. A method being able to run on mobile resources ensures data privacy and eliminates restrictions regarding user data management since all processing is happening locally on personal devices. However, should the framework be used in a social networking setting, additional measures should be taken in order to ensure that there are no sensitive data leaked through profile exchanges. All in all, the models developed in this paper provide a solid foundation for the incorporation of content aware features to image-based user modelling.

As future work, with a view to preserving the privacy of user profiles during the matching process, a promising avenue to explore is to encrypt and transmit the profiles and then calculate the matching score on the encrypted data using homomorphic encryption techniques [26].

Another extension worth considering is whether the deployed models can be dynamically updated on the user device. However, this is challenging as it is not straightforward to define what the training target would be and it would also cause different users to calculate their profiles with different models, which raises questions about how their profiles would then be compared.

ACKNOWLEDGMENT

This work has been supported by the HELIOS H2020 project, partially funded by the European Commission under contract number 825585.

REFERENCES

[1] D. Godoy and A. Amandi, “User profiling in personal information agents: a survey,” The Knowledge Engineering Review, vol. 20, no. 4, dec 2005.
[2] P. Wang, Y. Fu, H. Xiong, and X. Li, “Adversarial Substructured Representation Learning for Mobile User Profiling,” in Proceedings of the 35th ACM SIGKDD Conference. New York, NY, USA: ACM.
[3] H. Luo, C. Niu, R. Shen, and C. Ulrich, “A collaborative filtering framework based on both local user similarity and global user similarity,” in Machine Learning, vol. 72, no. 3. Springer, sep 2008, pp. 231–245.
[4] X. He, H. Zhang, M. Y. Kan, and T. S. Chu, “Fast matrix factorization for online recommendation with implicit feedback,” in Proceedings of the 39th Int. ACM SIGIR Conference on Research and Dev. in Information Retrieval. New York, NY, USA: ACM, 2016, pp. 549–558.
[5] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” Feb 2019.
[6] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youtube recommendations,” in RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems. New York, NY, USA: ACM, sep 2016, pp. 191–198.
[7] B. Thomee, B. Elizalde, D. A. Shamma, K. Ni, G. Friedland, D. Poland, D. Borth, Li, and L. Jia, “YFCC100M: The new data in multimedia research,” pp. 64–73, Feb 2016.
[8] P. Galopoulos, V. Gkatziazi, C. Iakovidou, S. Papadopoulos, and Y. Sotiriou, “PPIED2020 - Pinterest Interest Profiling Dataset 2020,” Jun. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3895162
[9] A. V. Savchenko, K. V. Demochkin, and I. S. Grechikhin, “User Preference Prediction in Visual Data on Mobile Devices,” Jul 2019. [Online]. Available: http://arxiv.org/abs/1907.04519
[10] S. Wiczorek, D. Filippaki, and A. Filipowska, “Semantic Image-Based Profiling of Users’ Interests with Neural Networks,” Tech. Rep.
[11] Q. You, S. Bhata, and J. Luo, “A picture tells a thousand words – about you! user interest profiling from user generated visual content,” 4 2015. [Online]. Available: http://arxiv.org/abs/1504.04558
[12] M. Cheung, J. She, and Z. Jie, “Connection discovery using big data of user shared images in social media,” 2015.
[13] M. Cheung, J. She, and X. Li, “Non-user generated annotation on user shared images for connection discovery:” IEEE, 12 2015, pp. 204–209.
[14] M. Cheung and J. She, “Detecting social signals in user-shared images for connection discovery using deep learning,” IEEE Transactions on Multimedia, vol. 22, pp. 407–420, 2 2020.
[15] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” vol. 2, 2006, pp. 1735–1742.
[16] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering.” IEEE Computer Society, 10 2015, pp. 815–823.
[17] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, “Deep metric learning with angular loss,” Proceedings of the IEEE International Conference on Computer Vision, pp. 2612–2620, 8 2017.
[18] W. Chen, X. Chen, J. Zhang, and K. Huang, “Beyond triplet loss: a deep quadruplet network for person re-identification,” Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 1320–1329, 4 2017.
[19] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, “Discriminative learning of deep convolutional feature point descriptors.” IEEE.
[20] R. M. Gray, Entropy and Information Theory, 2nd ed. Springer Publishing Company, Incorporated, 2011.
[21] M. Larson, M. Soleymani, G. Gravier, B. Ionescu, and G. J. Jones, “The MediaEval 2016 benchmarking initiative for multimedia evaluation: Mediaeval 2016,” vol. 24. IEEE Computer Society, 1 2017, pp. 93–96.
[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” vol. 23. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, pp. 4510–4520, 1 2018.
[23] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” 8th International Conference on Machine Learning, ICML 2019, pp. 10691–10700, 5 2019.
[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in CVPR09, 2009.
[25] Google, “Tensorflow lite — tensorflow,” 2019.
[26] C. Gentry, “Computing arbitrary functions of encrypted data,” Communications of the ACM, vol. 53, 3 2010.