T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

TAG RECOMMENDER FOR INSTAGRAM

TÜRKER TERCAN

SUPERVISOR
DR. GÖKHAN KAYA

GEBZE
2022

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

TAG RECOMMENDER FOR INSTAGRAM

TÜRKER TERCAN

SUPERVISOR
DR. GÖKHAN KAYA

2022
GEBZE

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 18/06/2022 by the following jury.

JURY

Member
(Supervisor) : Dr. Gökhan KAYA

Member : Assistant Professor Dr. Burcu Yılmaz

ABSTRACT

With the rising popularity of social media, **tags** or **hashtags** have come into our lives with their capability of a simple but strong and wide range of information. Users can share and annotate the image with any hashtags they like. These hashtags sometimes might be irrelevant or meaningless, which means the hashtag cannot be extracted or derived from the given image. We defined meaningful hashtags that can be extracted from the features from the given image. In this paper, we introduce a tag recommendation system for Instagram photographs that will recommend meaningful hashtags for Instagram social media application. It is really hard to construct such a dataset that each image is annotated with **multi-labeled** meaningful hashtags so, we used HARRISON dataset which has been contructed with real Instagram photographs and meaningless hashtags has been removed with post-processing. To extract visual features from images, we used deep convolutional neural networks (CNNs). We used two different **pre-trained** networks one of the trained on object based models and the other one is trained on scene-based model and used a **multi-label classifier** for HARRISON dataset. Finally, a simple web interface that enables users to upload and predict hashtags for the given image has been launched.

`https://turkertercan-hashtagrecommend.herokuapp.com/`

Keywords: Tags, Hashtags, Multi-label, Convolutional Neural Networks, Pre-trained networks, Multi-label classifier

ACKNOWLEDGEMENT

I'd like to express my gratitude to my supervisor, Dr. Gökhan KAYA, for contributing to this project and guiding me through our weekly sessions, as well as to Assistant Professor Dr. Burcu Yılmaz, for listening to me and guiding me through each meeting.

Additionally, during my education, I would like to express my thanks and devotion to my family, who provided me with unwavering support and knowledge, as well as to all of my professors who contributed to my development.

June 2022 **Türker Tercan**

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Tags or hashtags are widely used in social media applications such as Instagram, Twitter nowadays. The addition of the prefix '#' to a word is used to summarize the content of a user's post and draw the attention of other users or followers. Instagram application enables users to search for photographs that have been tagged with certain hashtags, therefore a hashtag may be used to increase a user's popularity. The majority of users annotate photographs with hashtags to increase their appearance, regardless of whether the hashtag is relevant to the image or not. As a result, the most of hashtags are used in a meaningless way. We defined a "Meaningful Hashtag" can be extracted or derived from the annotated image's features. Figure 1.1 shows meaningful and meaningless hashtags for Instagram.

#home, #hashtag, #follower, #pink, #makeup

#sea, #likeforlike, #followme, #sunday, #tagsforlikes

Figure 1.1: Example of meaningful and meaningless hashtags. Meaningless hashtags are colored as red.

With deep convolutional neural networks (**CNNs**), multi-label image annotation, image tagging, and multi-label classification have gained interest and success. In a variety of domains, including object classification, object detection[1], scene detection[2], [3], these fields have made significant breakthroughs. Regarding the variety of labels, image annotation[4] and hashtag suggestion task are similar. The labels of annotations consist mostly of surface information such as object in image and location

of image, while hashtags include inferential phrases, which need contextual awareness of images in addition to surface comprehension.

Addition to these methods, to annotate a image with multi meaningful hashtags, we used a word2vec skip-gram model and trained by hashtag text data dumped from HARRISON dataset's hashtag annotations. As a result, word embeddings for each hashtag is calculated. If some hashtags more used together, word embeddings' cosine difference will less which means these hashtags is often used together in a image. For example, assume that our hashtags are #smile, #fun, #tattoo. It is more likely to see #smile and #fun are used together more often then #smile and #tattoo. So, cosine distance between #smile and #fun hashtag word embedding will be less than #smile and #tattoo. Using hashtag word embedding and train them a CNN network, it'll help to predict more similar and meaningful hashtags instead of using one-shot vector.

In this paper, we introduce a method for giving "meaningful" tag recommendations for a given photograph.To do this, firstly, we used HARRISON[5] dataset, it has been constructed with real Instagram photograph and each photograph tagged with a variety of hashtags. We mentioned more details on next chapter2.3. Hashtags' word embeddings are extracted using Word2Vec Skip-gram model. After that, object features and scene features are extracted from two different pre-trained ResNet50 models to train a multi-label classifier. Trained multi-label classifier is used to predict real world Instagram images with a simple web interface, a Flask application. You can try our application with below url.

`https://turkertercan-hashtagrecommend.herokuapp.com/`

2. RELATED WORKS

2.1. Deep Convolutional Neural Networks (CNN)

Deep convolutional neural networks (CNN) has been successful in image recognition, object recognition and so on. More complex CNN architectures were introduced such as VGG Net[6], Google Net[7], Residual Net(ResNet)[8]. CNNs have proved its efficiency and success with applying this architecture to the many systems such as recommender systems, natural language processing and more. There is a bunch of pre-trained models are ready to use for any kind of application. Such as, ImageNet[9] which is trained for recognizing objects more than 20,000 categories and Places365[2] which is trained for recognizing scene descriptors with over 10 million images and 434 scene classes from an image.

2.2. Word2Vec - Skip-Gram

Word2Vec is one of the most popular technique to learn word embeddings using shallow neural network. It was developed by Tomas Mikolov in 2013 at Google[10]. Skip-gram allows us to convert any huge corpus to context-target word pairs. It produces pairs that are close to each other with a certain window size. So semantic information of each hashtag will be preserved. With context-target word pairs, word embedding's for each word can be produced. If the word pairs are often represented in word pairs, their cosine distance between them will be less.

We used a Word2Vec skip-gram algorithm because i want to make sure semantic information of hashtags are preserved. With this Word2Vec model for our hashtags, our training has been done with these word embeddings. So, more precise hashtags will be generated if their hashtags are commonly used together. We used to train my model both one-shot encodings and hashtags embeddings and, see which one is better.

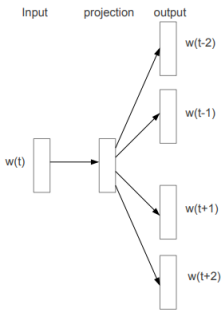

Figure 2.1: The Skip-gram model architecture from [10]. The training objective is to learn word vector representations that are good at predicting the nearby words.

2.3. Dataset - HARRISON

HARRISON[5] is a dataset constructed with real Instagram photographs. HARRISON satisfies the necessary conditions for our application. It contains 57,383 images and approximately 260,000 hashtags and each image has average of 4.5 hashtags. They have done post-processing with hashtags below hashtags are removed:

- Singular and plural form (#girl, #girls).
- Hashtags in the lower and upper case (#LOVE, #love).
- Hashtags in various forms of the same root word (#fun, #funny).
- Slang-inspired hashtags (#lol).
- Meaningless hashtags to gain the attention of followers (#like4like, #followforfollow).
- Hashtags are appear very less (#mancrush-sunday, #sightseeing).

illustration	art	architecture	ootd
drawing	coffee	hollywood	son
boyfriend	tired	vintage	boy
		design	instagood
		losangeles	love
		california	
		yellow	

Figure 2.2: Examples of the HARRISON[5] dataset, consisting of images and hashtags

3. ARCHITECTURE

3.1. Word2Vec

We've considered tag recommendation for Instagram images with three different steps. Firstly, we've trained Word2Vec model on HARRISON dataset hashtag text data, so it can extract the correlations between hashtags, we used such information to predict more semantically similar hashtags. Word embeddings of each hashtag is calculated. Then, we used those word embeddings of each hashtag in our loss function.

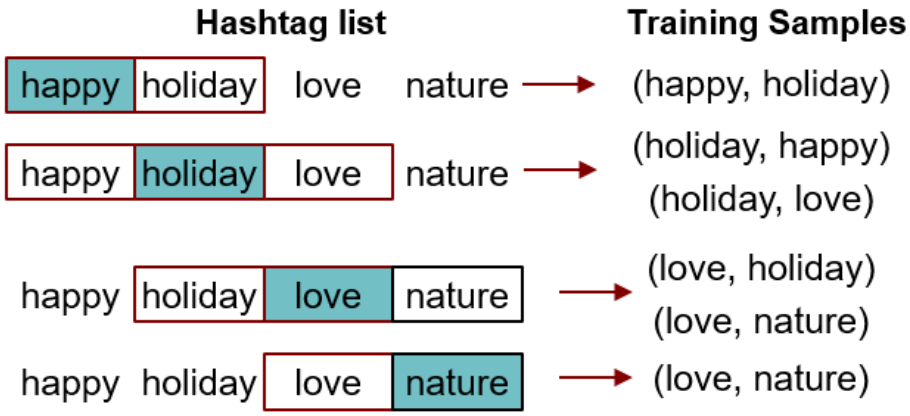

Figure 3.1: The word pairs produced with HARRISON dataset for our application.

3.2. Deep Convolutional Network Architecture

To extract features of images, we've used two pre-trained ResNet50[8] models, one of them we used for extracting object features from images which is called ImageNet[9]. The other one used for extracting scene features from images which is called Places365. You can see that feature extractor modules in Figure3.2 and Figure3.3.

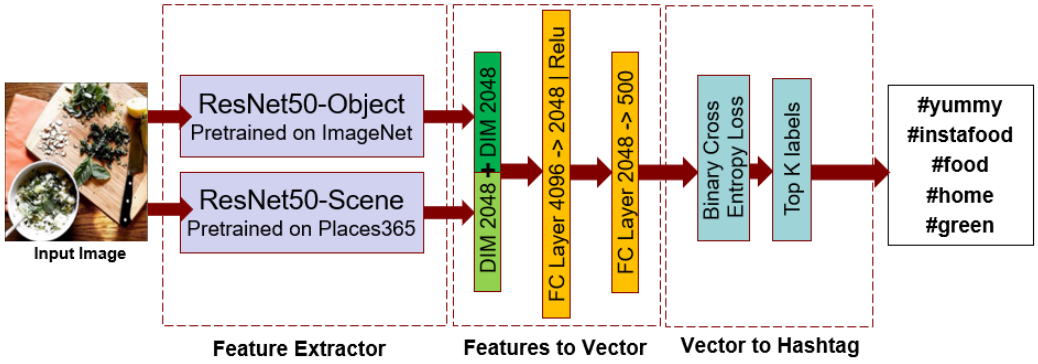

Figure 3.2: Overall first architectural design with concatenating two output features

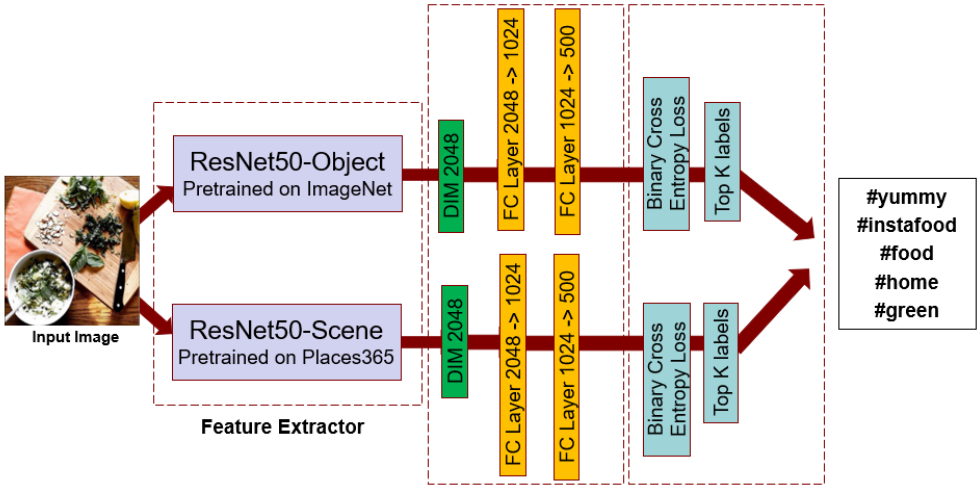

Figure 3.3: Overall second architectural design with separating two output features

We used two different designs on Features to Vector module. Firstly on Figure3.2, we concatenate two outputs of features, is used as the input of the classifier, and each fully connected layer of classifier have 2048 neurons. The output of this multi-label classifier is the word embeddings with size of vector 500. To recommend K number of hashtags, we sort the scores and extract the top K classes.

The difference between our first (3.2) and second (3.3) is we seperate two pre-trained classes, instead of concatenating the output features. We used two different multi-label classifier, to train our model and, to recommend K number of hashtags, we sort the scores of outputs of two multi-label classifier and extract the top $K/2$ classes and combine them together. If a class is in top $K/2$ classes in both outputs, scores are averaged.

We decided to go with second (3.3) architecture, since the metrics results and loss values are better than first (3.2) architecture.

3.3. Calculating Percentile Values

After we trained our multi-label classifiers, we needed convert scores of each classes to a certain percentage values. To do that, we normalized each output of images in test dataset and calculate standard derivation and mean for each hashtag. Let as assume our test samples are $D = \{x_i, h_i\}_i^M$ where x_i is the i-th image used as input of CNN, and Let $F(x)$ be the learned function of the deep CNN for the input image x, then the estimated labels y can be obtained by feeded the test input image x into the trained CNN model as

$$y = F(x)$$

The value of each element in the estimated vector y can be considered as the probability of the corresponding hashtag. Then we sort the scores of the y get top k hashtag classes. Let S_n scores samples of each hashtag and, size of hashtags is N. Let H_{ij}, C_{ij} is the top k scores and class indices of top k for each input test image $F(x_i)$.

$$S_n = [H_{0j}, H_{1j}, ..., H_{ij}] \text{ if } C_{ij} == n$$

After calculating all sample scores of each hashtag mean and standard deviation can be calculated. Let μ_n, σ_n is the mean and standard deviation of each hashtag and, M_n is the size of sample size of each hashtag.

$$\mu_n = \frac{\sum_{i=0}^{M_n} S_i}{M_n} \qquad \sigma_n = \sqrt{\frac{\sum_{i=0}^{M_n} (S_i - \mu_n)^2}{M_n}}$$

We saved those μ_n and σ_n values so that we can calculate the percentile value when predicting the images. Let assume submitted image X and, S_X and C_X is the

topk values of $F(X)$. We calculated Z-score for each class in C_X using S_X and convert it to percentile value. Let Z_i is the Z-score of each class in C_X and percentile value of each Z_i is P_i. After that, we used cumulative distribution function (CDF) from SciPy library to calculate percentile value.

$$Z_i = \frac{S_i - \mu_i}{\sigma_i} \qquad P_i = CDF(Z_i)$$

4. EXPERIMENTS AND EVALUATION

HARRISON[5] dataset has been constructed with real Instagram images. It has 57,383 images and approximately 260,000 hashtags. Each image has an average of 4.5 associated hashtags. For evaluation, we randomly split the dataset into 37,368 data for training and 16,015 data for validation. We've used two pre-trained ResNet50 models one of them trained on ImageNet[9] and the other one is trained on Places365[2] dataset.

4.1. Evaluation Measures

To compare the metric results with HARRISON[5] paper, we've decided to same evaluation measures from HARRISON. For each image, $Precision@K$ is defined as the portion of hashtags in top K ranked classes which match with the ground truth hashtags. $Recall@K$ is defined as the portion of hashtags in the ground truth hashtags which match with the top K ranked hashtags. $Accuracy@K$ is defined as 1 if at least one match between the top K ranked hashtags and the ground truth hashtags exists. Equations of the measures are shown as below where GT is ground truth hashtags:

$$Precision@K = \frac{|Result(K) \cap GT|}{|Result(K)|}$$

$$Recall@K = \frac{|Result(K) \cap GT|}{|GT|}$$

$$Accuracy@K = \begin{cases} 1 \text{ if } Result(K) \cap GT \neq \emptyset \\ 0 \text{ if } Result(K) \cap GT = \emptyset \end{cases}$$

4.2. Results

Table 4.1: The Performance of our architectural design compare to HARRISON performance

Architecture	**Metrics**		
	Precision @1	Recall @5	Accuracy @5
One multi-label classifier	28.21 %	19.54 %	43.87 %
Separate two multi-label classifiers	31.22 %	23.69 %	55.42 %
HARRISON architecture	30.16 %	21.38 %	52.52 %

Table 4.2: Comparison of Word2Vec and One-shot multi-vector on Seperate two-multi label classifiers architecture

Method	**Metrics**		
	Precision @1	Recall @5	Accuracy @5
One-shot multi vector	30.21 %	21.44 %	53.57 %
Word2Vec embeddings	31.22 %	23.69 %	55.42 %
HARRISON architecture	30.16 %	21.38 %	52.52 %

$Precision@1$, $Recall@5$, and $Accuracy@5$ is evaluated by averaging over all images in the test dataset. Table 4.1 shows evaluations results for our different architectures and HARRISON results. Compare to HARRISON. all three metric results our better in Separate two multi-label classifier architecture and has been shown best performance. So, scene-based features and object-based shouldn't be concatenated for a single multi-label classifier.

Table 4.2 has shown Word2Vec embeddings better performance than one-shot multi-vector. So that, Word2Vec hashtag word embeddings is working properly in our case since skip-gram[10] algorithm has produced word embeddings according to hashtags how many times used together.

Figure 4.1 - 4.9 demonstrates the example outputs and ground truth hashtags the test input images .Examples that have been shown below outputted from Word2Vec seperate multi-label classifier architecture. Simple hashtags are successfully recommended such as colors and objects.

Predicted Hashtags	Percentage Scores (%)	Ground Truth Hashtags
#pink	**100%**	**#heart**
#love	99.9%	**#pink**
#home	94.39%	
#heart	**84.13%**	
#cute	74.06%	
#happy	57.03%	
#family	53.52%	
#girl	49.95%	
#beautiful	47.23%	
#flower	44.91%	

Figure 4.1: Test Image 1 Result

In Fig 4.1 *#pink, #heart* are detected and present in ground truth hashtags. *#love, #home, #cute, #happy, #girl, #beautiful* not present in ground truth hashtags but still related hashtags according to the picture except *#family, #flower*,

Predicted Hashtags	Percentage Scores	Ground Truth Hashtags
#smile	**92.7**	#man
#happy	85.93	#boy
#instagood	82.9	**#guy**
#love	75.16	#model
#boy	**71.93**	**#smile**
#boyfriend	60.95	#spring
#new	56.35	#sun
#selfie	52	
#school	48.06	
#family	44.01	

Figure 4.2: Test Image 2 Result

In Fig 4.2 *#boy, #smile* hashtags are detected successfully. *#man, #model, #spring, #sun* hashtags couldn't detected but *#spring, #sun* ground truth hashtags are irrelevant to the given image. Predicted *#happy, #instagood, #boyfriend, #selfie* hashtags are relevant tags. *#new, #school, #family* hashtags are not relevant.

Predicted Hashtags	Percentage Scores	Ground Truth Hashtags
#beautiful	99.43	**#photo**
#photo	**99.41**	**#photographer**
#beauty	**98.96**	#song
#nature	81.04	#white
#vscocam	77.96	#flower
#photographer	73.51	#music
#instagood	73.07	**#beauty**
#love	70.06	#moment
#vsco	67.68	
#spring	64.72	

Figure 4.3: Test Image 3 Result

Fig 4.3 only just three matched hashtag is present which are *#photo, #photographer, #beauty*. We waited to see at least *#tree, #flower* hashtags to be predicted but the image contains more than 1 non-salient flowers maybe that's why our model couldn't predict them with object-features. In GT, *#song, #music* hashtags are imperceptible and *#nature, #love, #vsco, #spring* predicted hashtags are relevant to the image.

Predicted Hashtags	Percentage Scores	Ground Truth Hashtags
#animal	**100**	**#cat**
#winter	81.8	#catsofinstagram
#cat	**81.73**	**#tree**
#beautiful	78.46	#crazy
#tree	**76.77**	**#natural**
#nature	**74.19**	#love
#vsco	68.88	**#animal**
#bird	65.37	**#pet**
#black	62.56	#petstagram
#pet	59.43	

Figure 4.4: Test Image 4 Result

Fig 4.4 has successfully predicted *#cat, #tree, #natural, #animal, #pet* hashtags. *#crazy* hashtag ıs debatable in GT. Predicted *#winter* can be extracted from image since weather looks like cloudy. *#beautiful, #bird, #black* is miss predicted labels but since the cats sits on the tree object-model might have perceived as bird and the color of the tree and its branches the image are dark brown so that, scene-model might have produced *#black*.

Predicted Hashtags	Percentage Scores	Ground Truth Hashtags
#fashion	**100**	#zara
#style	**96.92**	#converse
#instagood	90.36	**#love**
#love	**62.03**	**#style**
#moda	53.07	#inspriration
#beautiful	48.79	#lifestyle
#black	46.4	**#fashion**
#girl	43.42	#blogger
#ootd	40.32	
#instafashion	40.01	

Figure 4.5: Test Image 5 Result

Fig 4.5, *#love, #style, #fashion* hashtags are predicted correctly. In GT, *#zara, #converse* hashtags are too specific to detect. In predicted hashtags *#black* is the only hashtags that not imperceptible from the test image.

Predicted Hashtags	Percentage Scores	Ground Truth Hashtags
#dinner	100	**#lunch**
#lunch	**99.96**	**#yummy**
#instafood	99.26	
#yummy	**98.77**	
#food	98.05	
#yummy	95.08	
#foodporn	85.26	
#foodie	83.04	
#instagood	81.63	
#delicious	78.5	

Figure 4.6: Test Image 6 Result

Fig 4.6, *#lunch, #yummy* hashtags are predicted correctly. Predicted hashtags are meaningful with good percentage scores.

Predicted Hashtags	Percentage Scores	Ground Truth Hashtags
#sunset	100	#nature
#sunrise	97.15	#naturelovers
#beach	90.11	#photo
#sun	77.17	#photooftheday
#beautiful	57.76	**#tree**
#tree	**45.04**	
#landscape	44.1	
#sky	38.21	
#sea	37.67	
#photography	36.16	

Figure 4.7: Test Image 7 Result

Fig 4.7, *#tree* is the only hashtag that is matched. Predicted *#sunset, #sunrise, #sun, #beautiful, #landscape, #sky, #photograph* hashtags is related but *#beach, #sea* hashtags are not related. The sand in the image might have seem as beach or sea for scene-model.

Predicted Hashtags	Percentage Scores	Ground Truth Hashtags
#happy	**100**	**happy**
#family	**100**	crazy
#smile	99.48	**family**
#school	86.94	
#love	84.86	
#friend	48.15	
#life	46.7	
#tired	45.7	
#teacher	44.5	
#funny	40.02	

Figure 4.8: Test Image 8 Result

Fig 4.8, *#happy, #family* hashtags are predicted correctly, but besides that the other hashtags *#funny* looks like imperceptible from the given image.

Predicted Hashtags	Percentage Scores	Ground Truth Hashtags
#ocean	100	**#beach**
#sea	**99.98**	#germany
#sand	98.43	#happy
#beachlife	91.3	#friend
#beach	**87.7**	**#sea**
#life	77.94	
#beautiful	74.91	
#island	73.64	
#wave	73.03	
#water	59.43	

Figure 4.9: Test Image 9 Result

Fig 4.9, model has successfully predicted *#beach, #sea* hashtags. In GT, *#germany, #happy, #friend* seems not relevant for this image, there is no clue about this image has shot in Germany or there is no one in the image. Predicted hashtags, *#ocean, #sand, #beachlife, #life, #beautiful, #island, #wave, #water* hashtags are perceptible from the given image.

Example results have shown good performance in terms of predicting hashtags that are simple and can be derived from image's content. It is not very good at detecting specific kind of objects such as *#converse* in Figure 4.5. Sometimes scenes can be difficult to distinguish between each other like in Figure 4.7 *#beach, #sea* or in Figure 4.9 *#ocean, #sea*. These problems be solved with increasing training data but in any case, there will be images that are annotated with not related hashtags. For example, Figure 4.3 *#song, #music*, or *#germany* in Fig 4.9.

4.3. User Tests

Will be updated

5. CONCLUSIONS

In this project, we developed tag recommender architecture for Instagram photographs. For our architectural design, we've used two different pre-trained ResNet50 models to extract object and scene features of the images, two separate multi-label classifiers and, trained images with their hashtags word embeddings. Results have shown good performance in terms of metric evaluation and predicting the images that are not from our dataset with good percentile scores. We've evaluated with three different metrics and two architectural design. Also, Word2Vec and one-shot multi-vector that represents hashtags vectors of each image are compared. We've used wide range of images with HARRISON dataset. System's hashtag prediction reliability should have increase with more training samples. Also, removing meaningless hashtags increase metric evaluations.

BIBLIOGRAPHY

[1] T. Lin, M. Maire, S. J. Belongie, *et al.*, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. arXiv: 1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.

[2] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[3] J. Shao, K. Kang, C. Change Loy, and X. Wang, "Deeply learned attributes for crowded scene understanding," Jun. 2015.

[4] V. N. Murthy, S. Maji, and R. Manmatha, "Automatic image annotation using deep learning representations," ICMR '15, pp. 603–606, 2015. [Online]. Available: https://doi.org/10.1145/2671188.2749391.

[5] M. Park, H. Li, and J. Kim, *Harrison: A benchmark on hashtag recommendation for real-world images in social networks*, 2016. eprint: arXiv:1605.05054.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[7] C. Szegedy, W. Liu, Y. Jia, *et al.*, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. arXiv: 1409.4842. [Online]. Available: http://arxiv.org/abs/1409.4842.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," pp. 248–255, 2009. DOI: 10.1109/CVPR.2009.5206848.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," vol. 26, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., 2013. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.