

Image tagging with an ensemble of deep convolutional neural networks

Marko Jovic*, Djordje Obradovic*, Vuk Malbasa*, Zora Konjovic**

* Faculty of Technical Sciences, University of Novi Sad, Serbia

** Singidunum University, Belgrade, Serbia

m.jovic@uns.ac.rs, obrad@uns.ac.rs, vmalbasa@uns.ac.rs, zkonjovic@singidunum.ac.rs

Abstract— We present a method for image tagging, i.e. assigning a set of tags/labels to an image. Three popular architectures of deep convolutional neural networks were used: VGG16, Inception-V3, and ResNet-50, which were pretrained on the ImageNet data set for a classification problem, and then fine-tuned on the HARRISON data set for the image tagging problem. The final model consists of an ensemble of these three convolutional neural networks, whose outputs were combined by different methods: averaging, voting, union, intersection and by two-layer feed-forward neural network. We verified these models on the hashtag recommendation for images from social network task, with a predefined set of 50 possible hashtags. All of the models were evaluated using the following metrics: precision, recall and F1-measure.

I. INTRODUCTION

In the field of computer vision, the number of studies on image understanding and visual analysis has recently grown significantly, with various works attempting to challenge topics such as object classification [1][2], object detection[3], scene classification [4], action recognition [5], image captioning [6], and even visual questing answering [7].

A related problem is image tagging, or hashtag recommendation, where the task is to predict a set of labels for an image, from a predefined set of labels. In contrast to the image classification problem, where the task is to predict exactly one class from given M classes (1 of M), image tagging is the task of predicting multiple classes (N of M , where N is arbitrary for every image). The image annotation task [8] is related to the image tagging task in regards to the diversity of labels. The labels and the related annotations mostly consist of directly apparent information such as objects in image and locations of image. However, hashtags include inferential words which require a contextual understanding of images. Additionally, a slew of words currently popular in social networks are also included in hashtags. We define a hashtag as any word attached to the prefix character '#' that is used in online social networks such as Facebook, Twitter, and Instagram. Hashtags are commonly used to summarize the content of a user's post, and attract the attention of other social network users.

On the Instagram site simple hashtags such as #dog and #beach describe the presence of simple objects or locations in a photo. Hashtags such as #happy or #sad express user's emotions, while abstract hashtags such as

#fashion and #spring specify topics. Inferential hashtags such as #colourful and #busy represent situational or contextual information. There are also advertising hashtags such as #likeforlike, which are not related to the photo's content. Since there are no rules for making and tagging hashtags, they can be diversely generated and freely used.

For example, there are simple cognate hashtags in both the singular and plural form (e.g. #girl, #girls), hashtags in the lower and upper case (e.g. #LOVE, #love), hashtags in various forms of the same root word (e.g. #fun, #funny), sentence-like hashtags (e.g. #iwantthissomuch, #kissme), slang-inspired hashtags (e.g. #lol), as well as meaningless hashtags to gain the attention of followers (e.g. #like4like, #followforfollow). Moreover, users can repeatedly tag the same hashtag for emphasis. Considering the wide variety and depth of context the recommendation of proper hashtags is a highly interesting and useful task in the age of social media [9].

Recommendation of hashtags has been studied in the fields of natural language processing (NLP), where some of the previous work focused on content similarity of Twitter text posts (tweets) [10][11], and unsupervised topic modeling with Latent Dirichlet Allocation (LDA) [12][13][14]. Although the significant progress in the field of of hashtag recommendation has been recently published by the NLP community, hashtag recommendation from images has not been strongly investigated in the field of image understanding. In a recent paper [15], the authors have presented a hashtag recommendation system, which uses the metadata of user information, such as gender and age, in addition to image data. However, if formulated as a vision-only problem where no metadata is used, the hashtag recommendation remains an open problem.

In this paper, we used deep convolutional neural networks (CNN), which have recently been proven to be a very useful tool in various computer vision problems, such as image classification [1], image captioning [6], image segmentation [16] and object detection [3]. With the introduction of multiple specialized frameworks for training deep neural networks on the GPU instead of the CPU, the time needed for training sophisticated architectures has dropped by orders of magnitude, making this whole area of research bloom.

To train and evaluate our model, we used the HARRISON data set [9], a benchmark data set for hashtag recommendation of real world images in social networks.

The HARRISON data set is a realistic data set, which provides actual images which were posted with associated hashtags on the online social network Instagram. The HARRISON data set has a total of 57,383 images and approximately 260,000 hashtags (with 997 unique hashtags). Each image has an average of 4.5 associated hashtags (minimum 1 and maximum 10 associated hashtags) due to the removal of infrequently used hashtags.

II. MODEL FOR IMAGE TAGGING

We propose a model for image tags prediction which has two levels: the first level consists of three different CNN architectures which are trained to predict tags for images, and on the second level we employed several different methods to combine the outputs from the first level models in order to improve the performance of the final prediction. Prior to training, all images are resized to 224x224, but the resizing is done by preserving the aspect ratio in order to maintain natural proportions of the objects in the image.

A. First level: fine-tuning pretrained CNNs

Although deep convolutional neural networks are currently *de facto* the standard as a tool for computer vision tasks, training them often requires a lot of training data (e.g. hundred thousands or millions of images) and plenty of resources (most state-of-the-art CNN architectures are trained for 1-3 weeks on multiple GPUs). However, it is possible to use so called *transfer learning*, i.e. using a model that is already trained to solve one problem to retrain it for another problem [17]. This procedure is usually called *fine-tuning* and it often dramatically reduces the needed number of training data and the training time.

Usually, this procedure is performed using already available, pretrained, CNN models on the ImageNet data set for the classification task. This data set consists of 1.2 million images, classified into 1000 different classes, and is used for benchmarking state-of-the-art of computer vision systems on the classification task. Since 2010, the ImageNet project has been running an annual contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [18], where researchers prepare software programs to compete to correctly classify and detect objects and scenes. In the deep learning era, winners of this contest publish their CNN architectures and corresponding parameters (neurons' weights) online, making them available to other researchers. We used several of these pretrained networks in this research.

It is important to note that, in order to fine-tune these CNNs, images for the new task need to be similar to the images in the ImageNet data set, i.e. they should be real-world images. For example, it is considered impractical to fine-tune ImageNet CNNs on medical images, since real-world images and medical images come from a completely different data generating process.

In this paper, we used three of the most popular architectures that competed on ImageNet ILSVRC:

Oxford's VGG16 [19], Google's Inception-V3 [20] and Microsoft's ResNet-50 [2]. Table 1 shows a brief comparison of these models.

TABLE I. – A brief comparison of three popular CNN architectures and their score on ImageNet LSVRC contest for classification task

Architecture	# of layers	# of parameters	ImageNet top-5 error (%)
VGG16	16	~140M	7.3
Inception-V3	42	~4M	6.7
ResNet-50	50	~0.85M	5.2

In order to fine-tune these models, we changed the number of neurons in the output layer from 1000 to 50, as we took only 50 most frequent labels (hashtags) to be relevant to our task, and then retrained them on HARRISON data set.

The most notable difference between the base models and fine-tuned models is in the activation function in the output layer – the pretrained models use a *softmax* activation function, whereas our fine-tuned models have a *sigmoid* activation function, which allows the fine-tuned models to be trained for the multilabel classification task.

B. Notation

Here, we introduce a notation to facilitate the explanation of the proposed model and different ensemble techniques.

Let x^i be i -th input image, where $i = \{1, 2, \dots, N\}$ and N is the size of the whole data set of images. Outputs from all models are represented as a vector $y^i = [y_1^i, y_2^i, \dots, y_M^i]^T$, where $y_j^i \in [0, 1]$ is an output for j -th label, and M is the total number of labels to predict, in our case $M = 50$. Outputs for the i -th image from the first level models will be denoted as $y_{VGG}^i, y_{INC}^i, y_{RES}^i$ for VGG16, Inception-V3 and ResNet-50 CNNs, respectively. An example of notation for output vector for first level model would look like $y_{VGG}^i = [y_{VGG,1}^i, y_{VGG,2}^i, \dots, y_{VGG,M}^i]^T$. We will use $y_{L1}^i = \{y_{VGG}^i, y_{INC}^i, y_{RES}^i\}$ to denote a set of outputs of all first level models for an image x^i .

The task of first level models is to learn a function $f: x \rightarrow y$, and thus output of first level models is $y_{L1}^i = f(x^i)$

The task of second level models is to learn a function $g: y_{L1} \rightarrow y$, and thus output of second level models is $y_{L2}^i = g(y_{L1}^i) = g(y_{VGG}^i, y_{INC}^i, y_{RES}^i)$, where $y_{L2}^i = \{y_{AVG}^i, y_{VOTE}^i, y_{UN}^i, y_{IN}^i, y_{FFNN}^i\}$ denotes a set of outputs of all second level models: averaging, voting, union, intersection and feed-forward neural network, respectively.

Since the input for second level models consists of outputs from first level models, the total number of inputs for second level models is $3 \times 50 = 150$.

C. Second level: ensembles

In order to increase the prediction accuracy of our model, we combined the outputs from the models in the first level using different approaches: averaging, voting, union, intersection and lastly, as a more sophisticated technique, a two-layer feed-forward neural network.

1) Averaging

For each label, output values from first level models are averaged.

$$y_{AVG,j}^i = \frac{y_{VGG,j}^i + y_{INC,j}^i + y_{RES,j}^i}{3} \quad (1)$$

2) Voting

For each label, output values from first level models are first rounded and then averaged.

$$y_{VOTE,j}^i = \frac{\text{round}(y_{VGG,j}^i) + \text{round}(y_{INC,j}^i) + \text{round}(y_{RES,j}^i)}{3} \quad (2)$$

3) Union

For each label to be positive, it suffices for at least one prediction of the first level model to be positive.

$$y_{UN,j}^i = \text{round}(y_{VGG,j}^i) + \text{round}(y_{INC,j}^i) + \text{round}(y_{RES,j}^i) \quad (3)$$

4) Intersection

For each label to be positive, prediction of all first levels model must be positive.

$$y_{IN,j}^i = \text{round}(y_{VGG,j}^i) * \text{round}(y_{INC,j}^i) * \text{round}(y_{RES,j}^i) \quad (4)$$

5) Feed-forward neural network

As a more advanced ensemble technique, we use simple two-layer feed-forward neural network, which has 150 input neurons, 150 neurons in the hidden layer, and 50 output neurons.

A diagram of the final model is shown in the Figure 1.

III. EXPERIMENTS AND RESULTS DISCUSSION

To quantify the performance of our model we choose the following metrics: precision, recall and F1-measure (as a harmonic mean of precision and recall). Since these

metrics only work with binary values {0,1}, we clipped and rounded the predictions of each of the models prior to evaluation.

First level models, i.e. pretrained CNNs, are fine-tuned by a stochastic gradient descent (SGD) optimizer with a learning rate of 1e-3, momentum 0.9, learning rate decay 1e-5, and batch size 32. Training of all CNNs was done on a single Titan X Maxwell GPU, and training each model took around 12 hours. The loss function we used consisted of two terms: binary cross-entropy and Kullback-Leibler divergence. As a form of regularization we performed data augmentation – all images were randomly rotated in the [-30, 30] degrees range, shifted horizontally and vertically in the [-10%, 10%] range, zoomed in the [-10%, 10%] range, and flipped horizontally.

The HARRISON data set was randomly split as 80% training data and 20% testing data. We should also note that, since we took only 50 most recent hashtags, this reduced the total number of images from 57,383 to 52,626. Since these models take a significant amount of time to train, our validation procedure consisted of validating against test data, which we resorted to in order to save resources. We used early stopping as a another form of regularization – F1-measure for test data is monitored each epoch, and if it doesn't improve for 10 epochs, the training of a model is stopped to prevent overfitting.

All second level models, except feed-forward neural network, don't need training, since their outputs are calculated directly. Two-layer feed-forward neural network is trained with the Adam optimizer with a learning rate of 1e-3 and a batch size of 512. For hidden neurons, we used *tanh* activation function. This neural network was trained in couple of seconds.

Table 2 shows evaluation results of first level models and the final models, depending on which ensemble was used. All results are averaged over all images in the test set. We can see that ResNet-50 model achieved the best results of all first level models, while feed-forward neural network outperformed other second level models.

Figure 2 demonstrates the examples of results for HARRISON data set in various cases. The shown examples are results of our best final model, i.e. two-layer feed-forward neural network ensemble.

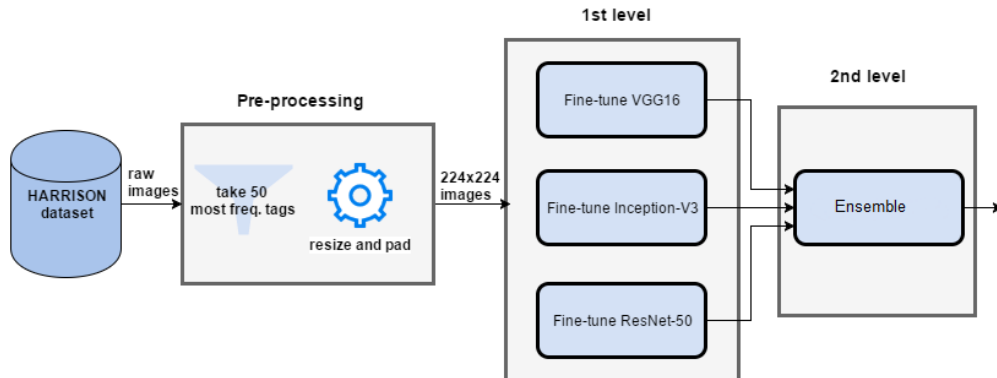


Figure 1 - A diagram of the final model

TABLE II. – Evaluation results for all models

Level	Model	Precision	Recall	F1-measure
1	VGG16	0.293	0.354	0.320
	Inception-V3	0.271	0.359	0.309
	ResNet-50	0.292	0.372	0.327
2	Ensemble – average	0.322	0.357	0.339
	Ensemble – vote	0.317	0.359	0.337
	Ensemble – union	0.231	0.467	0.309
	Ensemble – intersection	0.396	0.259	0.313
	Ensemble – FFNN	0.317	0.369	0.341










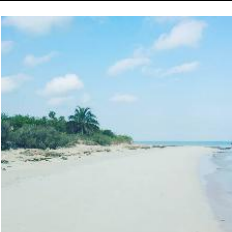
	TRUE funny PREDICTED friend, family		TRUE fashion PREDICTED art
	TRUE love, happy, selfie PREDICTED -		TRUE friend, snow, fun PREDICTED snow
	TRUE black PREDICTED fashion		TRUE yellow, flower PREDICTED yellow, home
	TRUE friend PREDICTED friend		TRUE dog PREDICTED snow, dog
	TRUE beach PREDICTED beach, sea		TRUE beach, beautiful, sun, nature PREDICTED beach, sea

Figure 2 – Examples of results for random images in test split of HARRISON data set

IV. CONCLUSION

In this paper we propose a model for automatic image tagging, which is organized in two levels: a first level that consists of three fine-tuned convolutional neural network architectures, and second model that is an ensemble of the first level models. Various ensemble approaches were discussed: averaging, voting, union, intersection and feed-forward neural network. These models were trained on HARRISON data set for hashtag recommendation and evaluated with three metrics: precision, recall and F1-measure. Our results suggest that ensemble with two-layer feed-forward neural network yielded the best results. However, it is important to note that it beat averaging ensemble by just a small margin, which means that one could choose an averaging ensemble over a feed-forward neural network ensemble in order to save resources. With respect to HARRISON data set, our results show that hashtag recommendation task is highly challenging due to the difficulty to understand contextual information and inferring of the user's intent.

Future directions of research could be improving the first level models with more intensive hyper-parameter optimization. Also, various other machine learning classifiers could be employed as second level models. If resources allow it, performance may further be improved with third level models, which would take the output of the second level models as an input. It is important to note that our models ignore the dependencies between image tags since we considered hashtags as independent labels for training our multi-label classifiers. This makes combining with NLP techniques such as word similarity also an option to improve our models.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, p. 2012.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *ArXiv Prepr. ArXiv151203385*, 2015.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [4] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.
- [5] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4305–4314.
- [6] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," *ArXiv Prepr. ArXiv150203044*, vol. 2, no. 3, p. 5, 2015.
- [7] H. Xu and K. Saenko, "Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering," *ArXiv Prepr. ArXiv151105234*, 2015.
- [8] V. N. Murthy, S. Maji, and R. Manmatha, "Automatic image annotation using deep learning representations," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 603–606.
- [9] M. Park, H. Li, and J. Kim, "HARRISON: A Benchmark on HAShtag Recommendation for Real-world Images in Social Networks," *ArXiv Prepr. ArXiv160505054*, 2016.
- [10] T. Li, Y. Wu, and Y. Zhang, "Twitter hash tag prediction algorithm," in *ICOMP'11-The 2011 International Conference on Internet Computing*, 2011.
- [11] E. Zangerle, W. Gassler, and G. Specht, "Recommending #-tags in twitter," in *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, 2011, vol. 730, pp. 67–78.
- [12] F. Godin, V. Slavkovicj, W. De Neve, B. Schrauwen, and R. Van de Walle, "Using Topic Models for Twitter Hashtag Recommendation," in *Proceedings of the 22Nd International Conference on World Wide Web*, New York, NY, USA, 2013, pp. 593–596.
- [13] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," 2009, p. 61.
- [14] J. She and L. Chen, "TOMOHA: TOPic model-based HAShtag recommendation on twitter," 2014, pp. 371–372.
- [15] E. Denton, J. Weston, M. Paluri, L. Bourdev, and R. Fergus, "User Conditional Hashtag Prediction for Images," 2015, pp. 1731–1740.
- [16] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *CoRR*, vol. abs/1605.06211, 2016.
- [17] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *CoRR*, vol. abs/1411.1792, 2014.
- [18] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis. IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
- [19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014.
- [20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *CoRR*, vol. abs/1512.00567, 2015.