Abstract Concept & Emotion Detection in Tagged Images with CNNs

Youssef Ahres, Nikolaus Volk
Stanford University
Stanford, California
yahres@stanford.edu, nvolk@stanford.edu

Abstract

Common computer vision classification models try to classify images into objective object categories. Rather than object classification, the goal of this paper is to learn and detect abstract concepts and emotions in images using FLICKR images and their tags. The baseline model is a VGG-16 Convolutional Neural Network (CNN) which outputs binary predictions for each single concept. Furthermore, we present and evaluate two different methods to deal with highly skewed data, a common problem in such specific classification tasks. In addition to the classic cost weighting, we propose a novel approach using entropy-based mini batch sampling. Experimentally, we explore the ability of our CNN model to learn these concepts. We also show that our entropy-based mini batch model outperforms the baseline and the model with modified weights, using F1-score metrics. Finally, we investigate the tag noise level to further detail our quantitative results.

1. Introduction

Most computer vision models try to classify and recognize an image without its surrounding (textual) context and focus mainly on classifying defined object categories (car, cat,...). Specifically in the context of social media, this leads to a significant loss of information, particularly when thinking about all the hashtags / tags that are used to give a particular image post a specific sentiment and meaning. E.g. instead of just posting an image with a cat, the social media user would post this image together with tags such as "$\#cute$ kitty, $\#beautiful$, $\#weekend$ with my puppy". Other example images together with their tags are shown on Figure 1.

The goal of this project is to learn and detect conceptual information using tags as labels for concepts using convolutional neural networks (CNNs). Specifically, given an image we are predicting whether a certain concept / emotion out of a pre-defined list of concepts is contained in the image.

Potential areas of application include social media profiling, image sentiment analysis and image search. Ultimately combining high level text semantic extraction with a powerful visual object- & concept-classification framework will be of high future interest to understand complex textual-visual documents & media in the field of information retrieval.

One of the main challenges encountered is our sparse data set. Because, each concept is only contained in a small fraction of all images, concept labels highly skewed towards the 0-class. Therefore, the initial baseline model tends to have low detection recall. A lot of effort has been dedicated to overcome this challenge and this paper summarizes the approaches developed as well as the results obtained.

The rest of this paper is organized as follows: we first review the existing related literature and compare it to our specific task. Then, we present the data set with examples and the specific related challenges. Third, we describe the baseline model and present 2 different approaches to deal with highly imbalanced data. Fourth, we present our results analyzing the model performance on a per-tag basis as well as the general data imbalance approaches. Furthermore, we will investigate the noise level within our experiments as a direct consequence of the very 'subjective' nature of using tags rather than using descriptive annotations. Finally, we conclude by discussing further research opportunities and challenges in this area.

2. Related Work

Over the past decade models using CNNs in computer vision continuously pushed performance boundaries in classification tasks such as the large scale visual recognition ImageNet challenges [1]. The most common discipline here is object classification with its direct relevance for various applications in industry such as autonomous driving or image search.

In recent years, hand in hand with the advances of deep learning in Natural Language Processing (NLP) research has been growing in the field of multimodal learning: combining visual and textual information. One common field of

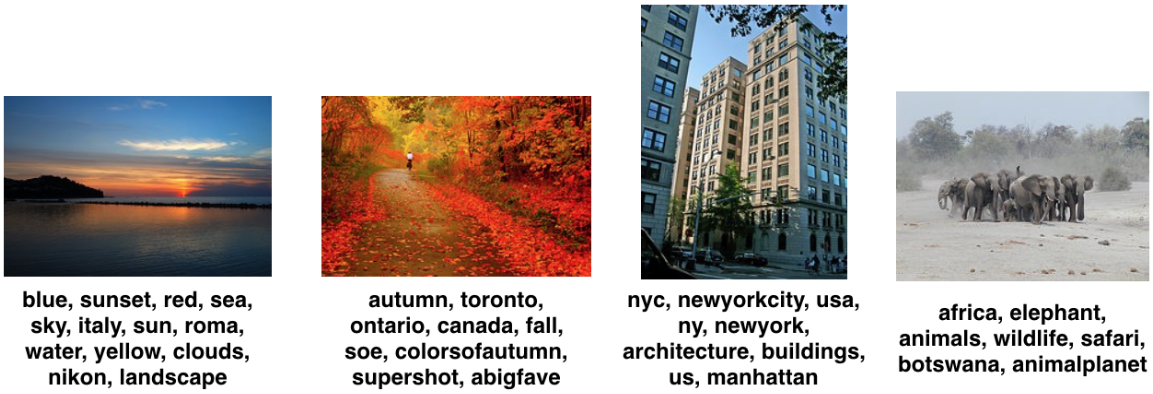

blue, sunset, red, sea, sky, italy, sun, roma, water, yellow, clouds, nikon, landscape
autumn, toronto, ontario, canada, fall, soe, colorsofautumn, supershot, abigfave
nyc, newyorkcity, usa, ny, newyork, architecture, buildings, us, manhattan
africa, elephant, animals, wildlife, safari, botswana, animalplanet

Figure 1. Example images and their corresponding tags from the NUS Wide data set.

application is provided by social media where images are mostly embedded in a textual context.

Chen et al. [2] as well as Xu et al. [3] focus on visual sentiment analysis tasks, similar to sentiment analysis in NLP applications. They are incorporating tag sentiments into the image classification pipeline. Other models try to include comments or other social network metadata, mostly using graphs, ranking approaches and complex image-text pipelines (e.g. [4] or [5]).

Other models such as [6] relate semantics found in tags and image semantics in a common representational vector space and are able to provide a common search space between tags and images.

Most of these papers lack a general approach to learn, predict and detect abstract concepts without a supporting NLP pipeline. Besides that, the question what concepts or emotions are actually ’learnable’ to which extent remains largely unaddressed.

Besides exploring the field of learning abstract concepts and emotions we are dealing with the issue of a highly imbalanced data set for each concept. A straightforward method to deal with highly biased data is sub-sampling the majority class or duplicating the minority class(es) [7]. However, this approach is limited for multi-label classification problems as we are dealing with as duplicating one minority class would increase another majority class which would result in counterproductive results.

Another common approach to address this issue that we will explore our baseline to is training a ’cost-sensitive’ model [8] [9]. Other methods use ensemble methods such as different SVMs [10] to overcome the class imbalance.

We propose propose a novel method in sampling mini batches using an information theory approach. Maximizing information gain for sampling is commonly used in the area of selecting the most informative samples when data labeling is expensive, called active learning [11]. We see this new method as one of the main contributions of this paper in computer vision / deep learning tasks.

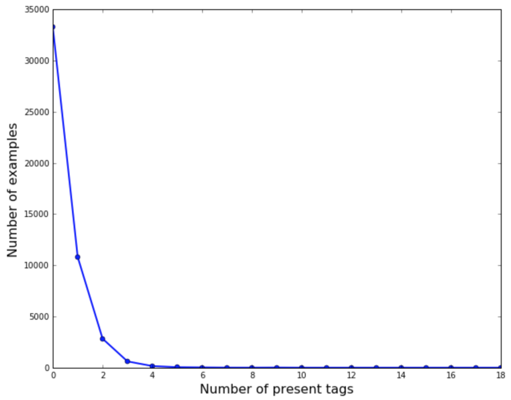


Figure 2. Distribution of the tags in the data set

3. Dataset

We used the publicly available dataset NUS-WIDE [12] which contains 27,000 Flickr ($https : //www.flickr.com/$) images and additional tags for each image (appr. 4000 unique tags). Tags can be descriptions of the image such as *landscape*, indicating that the image is picture of a landscape. Or it could be more about the author such as *abigave*, a seemingly popular Flickr user. Since the objective of our task is to learn high level concepts, we are more interested in the former type of tags. Examples of the images with their respective tags are shown in Figure 1.

The first step is to select the tags we will learn. We computed the frequency of the tags in the data set to select frequent tags that also include interesting concepts to learn. Surprisingly, many of the most frequent tags, including the most frequent one, *abigave*, refer to Flickr users or groups. Based on this tag-frequency analysis, we narrowed

Tag	# examples	Coverage
Landscape	1527	13.5%
Wildlife	591	5.2%
Travel	1036	9.1%
Vacation	476	4.2%
Sunrise	412	3.6%
Sunset	1486	13.1%
Night	1047	9.3%
Art	1224	10.8%
Architecture	1200	10.6%
Urban	707	6.25%
Abandoned	339	3%
Beautiful	711	6.3%
Cute	508	4.5%
Love	489	4.3%
Beauty	423	3.8%
Summer	786	6.9%
Fall	977	8.7%
Winter	727	6.5%
Spring	554	5%

Table 1. Summary of the 19 tags of interest: first column is the tag text, the second contains the number of example for the tag and the third its frequency in the filtered and final data set

our number of tags to learn to 19 tags using a hard threshold of 300 examples for every one of them. The final tags are shown in Table 1.

Despite our efforts to only select frequent tags to learn, most of the images in the dataset do not contain any. Figure 2 shows how many images in the data set have tags in them. The x-axis refers to the number of present tags out of the 19 tags of interest. The y-axis shows the corresponding number of example images in the data set.

Given the sparsity of this data set, we filter out the images which do not contain any of our 19 concepts of interests and obtain a training data set with 11317 images and a validation data set with 3178 images. On a per-concept basis, our data is still very sparse and skewed towards the 0-class (concept not contained). Table 1 shows the final tags selection as well as their respective number of examples and frequency in the data set.

4. Methodology

We approach this problem as a multi-label classification problem. Every tag is detected using a logistic regression. All these classifiers are built on top of a shared CNN. As we detail below, the CNN extracts the features from a given image and feeds them to 19 binary classifiers to detect whether the associated tag is present or not.

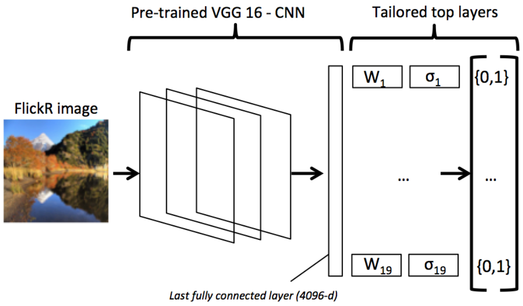


Figure 3. Structure of our learning model: a shared VGGNET-16 with 19 different logistic regression classifiers, one for each tag

4.1. Baseline Model: Finetuned VGGNET-16

Our baseline model is a VGGNET-16 [13] with 19 different logistic regression classifiers instead of the last softmax layer. Each of these classifiers corresponds to one of our tags and predicts its presence in the image. The overall structure is displayed on Figure 3. Formally, the top layers we added to the VGGNET-16 have the following equations:

$$L_i = \frac{1}{1 + e^{-\theta_i^T x}}$$

for $i \in landscape, wildlife, travel....$ x refers to the the features outputted by the previous fully connected layer of size 4096. And θ_i refers to the weights of these features for the tag i.

Because of the skewness of our training data, this baseline model performed poorly overall as we will discuss in the results section. Therefore, two methods were developed and implemented to overcome the data imbalance issue.

4.2. Introducing Class Weights

The first approach to overcome class imbalance in the data is to modify the cost function directly in the model. Various papers such as [8] or [9] explored the implementation of such 'cost-sensitive' models.

The cost in this case is multiplied by a weighting term w_1 for the non-zero classes, and $w_0 = 1 - w_1$ respectively for the 0 class. The modified cross entropy cost function for one aspect therefore results in:

$$E_a = \sum_i w_i y_i log(\hat{y}_i) \quad (1)$$

with the following constraints:

$$\sum_i w_i = 1 \quad (2)$$

Intuitively, we aim to penalize the cost more when we miss existing tags, forcing the model to increase tag-recall (or tag-sensivity which describes the same). If the weight assigned to w_1 is smaller than w_0, the resulting model will predicts only zeros as it will give the majority class an even higher weight. On the other extreme, if w_1 is much larger than w_0, our false positive rate can potentially be harmful. Therefore, we need to be careful when setting these hyperparameters.

We select the weights based on the inverse frequency of the tag. We also add a smoothing parameter k to avoid putting too much weight on the minority class.

$$w_{landscape=1} = \frac{n_{landscape} + k}{total_{training} + k} \tag{3}$$

$$w_{landscape=0} = 1 - w_{landscape=1} \tag{4}$$

Following what appears to be a common practice [14], we set k such as none of the classes weights more than the double of the majority classes. We implemented this method directly in Caffe Softmax layer by adding an optional argument in the proto buffer and adding the adequate equation in CPU as well as GPU forward/backward propagation. We are currently starting an effort to push the result for code review and contribute to the Caffe project.

4.3. Entropy-enriched minibatch sampling

We propose a novel approach to solve this problem. This heuristic method is based on information-rich mini-batches. Maximizing information gain for sampling is commonly used in the area of selecting the most informative samples when data labeling is expensive, called active learning [11, 15]. We are showing that this general idea can be used in the area of minibatch sampling.

Each mini batch is created using the following procedure:

1. Randomly sample candidates (examples) from the training set
2. Choose the candidate that maximizes the entropy of the current mini batch.

Entropy is the expected value of information contained in a set. Therefore, maximizing this metric is likely to maximize the information capability of each batch.

Formally, the entropy is defined as follows:

$$H(X) = \sum_{i \in C} P(x_i) I(x_i) = -\sum_{i \in C} P(x_i) log_2(P(x_i)) \tag{5}$$

Where C defines the set of classes, here a vector of length 19 where every index represents a tag. And the probabilities are computed using the following:

$$P(x_i) = \frac{n_i}{\sum_{j \in C} n_j} \tag{6}$$

where n_i is the number of times the class (tags) i appeared in the data. However, when building the mini batches, we start with an empty set. Therefore, initially all probabilities are 0 and the entropy is not defined unless all the classes are represented at least once. Following the convention on entropy calculation, we compute entropy only on classes that are already present in the batch. Adding a new class will always have a higher information gain than adding a sample to an existing class. This will force the batch selection to be as diverse as possible.

```
   Data: $Y$,$n_{candidate}$,$size$,$n_{batches}$,$replacement$
   Result: $batches$
1  $batches \leftarrow EmptyList()$;
2  for $i = 1..n_{batches}$ do
3      $c\_batch \leftarrow EmptyList()$
4      for $j = 1..size$ do
5          $candidates \leftarrow sample(Y, n_{candidate})$
6          $best \leftarrow choose\_best(candidates, c\_batch)$
7          $c\_batch.add(best)$
8          if $replacement = False$ then
9              $discard(Y, best)$
10         end
11     end
12     $batches.add(c\_batch)$
13 end
```

Algorithm 1: Create Entropy-based mini batches

```
Data: $candidates$,$c\_batch$
Result: $best$
$entropies = EmptyDictionary()$
forall $candidate \in candidates$ do
    $entropies[candidate] \leftarrow$
      $computeEntropy(c\_batch + candidate)$
end
$best \leftarrow argmax(entropies)$
```

Algorithm 2: *choose_best* procedure

Algorithm 1 shows how we create these batches based on the data: The inner loop (line 4) summarizes the creation of a single batch: we select an arbitrary number of candidates (line 5) to add the mini batch, generally between 5 and 10. We find the best candidate using Algorithm 2 (line 6) and add it to the batch. Note that this algorithm works with and without replacement.

Algorithm 2 shows the procedure we use to find the best candidate for the current mini batch: We will compute the entropy of the current batch *c_batch* with every candidate in the set. This process is analog to the information gain measure. If some classes are not present in the batch, we will compute the entropy based on the present tags only. Finally, once we compute the entropy of the current batch with all candidates, we simply choose the one that maximizes it.

Figure 4 shows the the diversity in a batch using this method and a regular random minibatch selection. We see

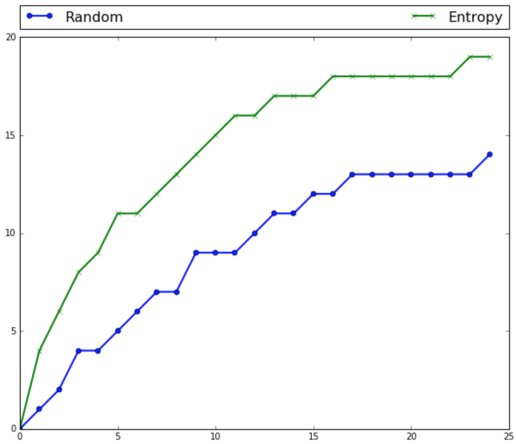


Figure 4. Effect of entropy sampling on batch diversity. The x-axis shows the batch size and the y-axis the number tags present in the batch

that, following our intuition, the minibatches created using the entropy-based sampling quickly achieve a higher diversity and are able to catch all 19 tags within the 24-sized batches we have. We selected 24 because it was the maximum the AWS machine was able to process before running out of memory.

5. Experiments

5.1. Implementation

Due to our limited data size we choose to use a pretrained VGG model from the Caffe model-zoo. This model has pretrained weights which will be used to initialize our model. The VGG-16 net was trained on ILSVRC-2012 [16] and achieved 13.1 error rate in the top 5 ImageNet challenge. Backpropagation will take place both in our customized top layers as well as through the pretrained layers.

With respect to the infrastructure, we used a GPU-powered AWS-EC2 instance with Caffe and Pycaffe.

Caffe requires either dumping image data into special data formats (LMDB or HDF5) or constructing special data layers. As LMDB only allows one label per image (common classification problem), we select to use HDF5. Data had to be processed in batches to fit in our limited memory. In addition, we subtract the mean image (calculated over the training set) and reformat all images to be 224x224x3 in the data layer.

As the model differs significantly from the Vanilla VGG-16 Classification Net, changes to the solver and learning parameters were made. Adagrad turns out to be the most favorable learning policy. The base learning rate is cross-validated to be 10^{-4}. Finally, the batch size was 24, the maximum we were able to use for the AWS GPU that has only 4GB of memory.

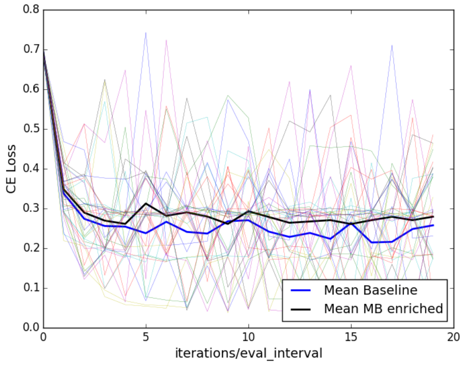
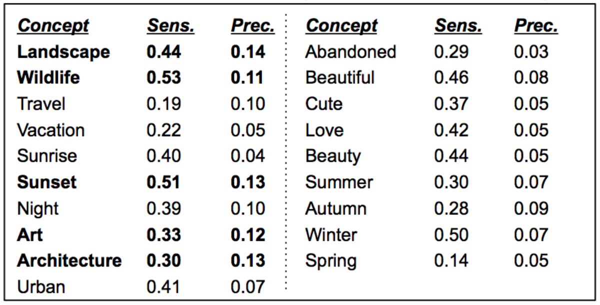


Figure 5. Losses of baseline and mini batch enriched model



Concept	*Sens.*	*Prec.*	*Concept*	*Sens.*	*Prec.*
Landscape	**0.44**	**0.14**	Abandoned	0.29	0.03
Wildlife	**0.53**	**0.11**	Beautiful	0.46	0.08
Travel	0.19	0.10	Cute	0.37	0.05
Vacation	0.22	0.05	Love	0.42	0.05
Sunrise	0.40	0.04	Beauty	0.44	0.05
Sunset	**0.51**	**0.13**	Summer	0.30	0.07
Night	0.39	0.10	Autumn	0.28	0.09
Art	**0.33**	**0.12**	Winter	0.50	0.07
Architecture	**0.30**	**0.13**	Spring	0.14	0.05
Urban	0.41	0.07			

Figure 6. Sensitivity & Precision for labels with MB enriched model

5.2. Results

First, we note the difficulty of the task at hand. Figure 5 shows the learning curves for the baseline model and the entropy-enriched minibatch sampling which stays at a higher loss as the minibatches contain more difficult information. The weighted model results in a very similar curve than the baseline model. Even if the mean is relatively smooth, we see that the curves per label are very noisy. This is partly due to the lack of training example per tag in every batch. Since the batch size is only 24 and we have 19 tags, most of the batches contain at most two examples for every tag. The learning is therefore very noisy.

Second, our experiments showed that the models capability to learn varies strongly depending on the concept. One can clearly see a correlation of the 'abstractness' and performance. Or in other words: Concepts that are more 'object-related' ($landscape, wildlife, ...$) are easier to detect in general, concepts that are very abstract ($beauty, cute, ...$) are harder to detect. Another clear correlation is that concepts with more positive training examples result in better performance.

Figure 6 shows our quantitative results of the best model.

In order to analyze and compare our approaches, we identify and highlight the top performing (top 5) tags that the model can learn best, using the well-known F1-score

Model	F1-Score
Baseline	0.16
Weighted	0.14
Entropy enriched	0.18

Table 2. Model performance

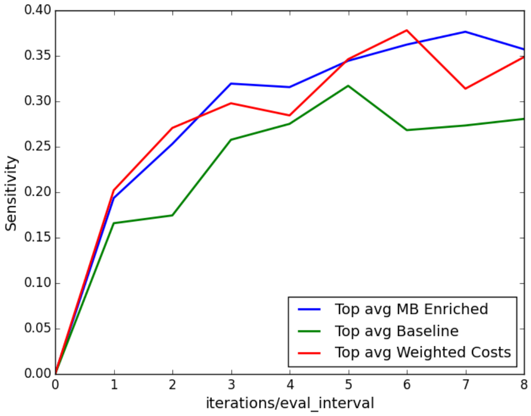


Figure 7. Sensitivity progress progress for baseline & modified models with avg. over top 5 concepts

(harmonic mean of sensitivity & precision). These tags are highlighted in bold on Figure 6. Taking the mean value of the top 5 tags, we now evaluate the performance of our 3 models.

We start the comparison between our approaches using the F-1 metric on the top 5 tags. The result is shown in Table 2. Our entropy-enriched minibatch model outperforms the baseline and the weighted model. Even though the weighted model has good sensitivity performance, it lacks precision which makes sense intuitively as it puts very high importance on detecting the 1-class.

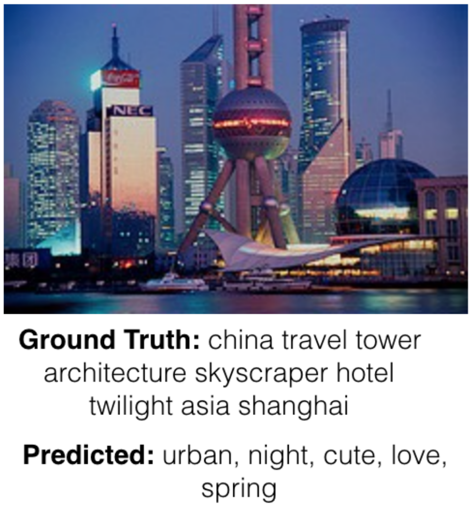
Figure 7 shows the sensitivity versus iteration number for these high performing tags.

6. Discussion

6.1. Tags Discrepancy

As we pointed out in the previous section, some tags tend to perform much better than others. Highly abstract concepts are not properly learned because they require a lot more inference and the information present in the image might not be enough. For instance, the $spring$ tag just refers to the date the picture was taken with no real evidence on the image that it is the spring. On the other hand, $winter$ outperforms it because most the $winter$-tagged image contain snow and ski trips photos.

On the other side of the performance spectrum, some tags tend to perform relatively well. In particular, the $landscape$ tag seems to have high recall. The first example


Ground Truth: china travel tower architecture skyscraper hotel twilight asia shanghai

Predicted: urban, night, cute, love, spring

Figure 8. Example of NUS WIDE image with ground truth and predicted labels

image on Figure 1 shows a typical landscape image. These images are easily recognizable by a human observer as opposed to $cute$ where there is no typical image. This explains the general discrepancy between high performing and low performing tags.

The number of examples for a tag is another indicator for this discrepancy. Obviously, more examples for a given tags improves its prediction performance. For instance, $landscape$ and $sunset$ are the tags with the largest number of examples (appr. 1500) and they tend to perform much better than $abandoned$ that has only appr. 300 examples.

6.2. Noisy Data

Another challenge of this work relates to the level of noise in the data. Tags do not necessarily describe the image and even when they do so, people use different vocabulary to express the same thing: for instance people can use $urban$ or $city$ for the exact same image. We preprocessed the tags to include synonyms such as $fall$ and $autumn$ in the same tag. However, we weren't able to map all tags. This tends to hinder our quantitative results even if the model performs relatively well. For instance, Figure 8 shows an image of Shanghai at night along with its ground truth and predicted tags. We see that the ground truth does not contain $night$ nor $urban$, which could be good tags for this image. The model seems to catch them along with other low-performing tags. In this case, it would count as a false positive for both $urban$ and $night$ tags decreasing the precision. That does not imply that the model for these tags is very accurate, but the level of noise in the data seems to be very high and therefore explains, at least partly, the low precision numbers.

6.3. Class Imbalance

With regards to the strategies used to deal with the class imbalance in the data, weighting the cost function is not able to overcome this issue for our model. We assume that weighting the cost function requires more extensive cross validation of parameters to add value to the learning process. However, the entropy-sampled method outperforms significantly the baseline (and the weighted method). Intuitively, this points out the opposite nature of these techniques: the weighted cost function tends to force the classifier to choose minority classes while the entropy-sampling drastically balances the training set.

Specifically using different sampling parameters (called $n_{candidate}$ in Algorithm 1) indicating the number of examples the mini batch sampling method should sample from gives more insights when and why the mini batch sampling works best: A high parameter (> 10) gives too much flexibility to the method and allows it to choose from a small subset of the training data samples (which increase the entropy but which are not a good representation of the overall data). With a low parameter (< 3) the method is not able to find good examples. A good trade-off lies around $n_{candidate} = 5$ where the model has a good balance between diversity and entropy-enrichment. Further work should explore ways to automatically choose this parameter.

7. Conclusion

In this paper, we compared various approaches to tackle visual abstract concept detection. The tagged images add two layers of complexity: The issue with very abstract and conceptual noisy tags, often in the form of subjective emotions and the issue of class imbalance for each concept.

With respect to the first issue we can conclude that, as discussed in section 6.1 and 6.2, tags should not be considered as ground truth labels, but rather as a more loose version of 'concept indicators'. We showed that concept learning strongly depends on the type of concept and compared the ability to learn these different types of concepts.

With respect to the high class imbalance, we conclude that a highly biased data set requires creative solutions. Therefore, we proposed and implemented two extensions to overcome these challenges including a novel method based on information gain which actually outperforms the baseline model. We assume that weighting the cost function is a suitable approach in general as shown in related work but requires more extensive cross validation. Our novel concept of entropy-based minibatch sampling seems very suitable to deal with highly biased datasets.

We conclude by outlining a few areas for future work. We mainly see the entropy-based sampling as an intelligent sampling approach that seems very promising to explore further. We propose investigating the effects of the entropy-based sampling on the learning process and how it affects accuracy on different visual classification problems. Using the presented results, one may find an even better way to optimize the training pattern for a more balanced classifier. In addition, it would be fruitful to investigate efficient techniques to optimize its hyper-parameters such as number of candidates or batch size.

References

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[2] Tao Chen, Damian Borth, Trevor Darrell, and Shih-Fu Chang. Deepsentibank: Visual sentiment concept classification with deep convolutional neural networks. *arXiv preprint arXiv:1410.8586*, 2014.

[3] Can Xu, Suleyman Cetintas, Kuang-Chih Lee, and Li-Jia Li. Visual sentiment prediction with deep convolutional neural networks. *arXiv preprint arXiv:1411.5731*, 2014.

[4] Elia Bruni, Giang Binh Tran, and Marco Baroni. Distributional semantics from text and images. In *Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics*, pages 22–32. Association for Computational Linguistics, 2011.

[5] Jinhui Tang, Shuicheng Yan, Richang Hong, Guo-Jun Qi, and Tat-Seng Chua. Inferring semantic concepts from community-contributed images and noisy tags. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 223–232. ACM, 2009.

[6] Yue Gao, Meng Wang, Zheng-Jun Zha, Jialie Shen, Xuelong Li, and Xindong Wu. Visual-textual joint relevance learning for tag-based social image search. *Image Processing, IEEE Transactions on*, 22(1):363–376, 2013.

[7] Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1), 2004.

[8] Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

[9] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *Knowledge and Data Engineering, IEEE Transactions on*, 18(1):63–77, 2006.

[10] Rong Yan, Yan Liu, Rong Jin, and Alex Hauptmann. On predicting rare classes with svm ensembles in scene classification. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–21. IEEE, 2003.

[11] Rita Chattopadhyay, Zheng Wang, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Batch mode active sampling based on marginal probability distribution matching. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):13, 2013.

[12] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, page 48. ACM, 2009.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Simon N Wood. Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(2):413–428, 2000.

[15] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[16] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.