Social Image Tag Relevance Learning based on Pixel Voting

Wei Liu¹, Yunxing Ruan¹, Xia Cai¹ and Hao Chen²*
¹Computer School, Central China Normal University, Wuhan 430079, China
²School of Engineering and Architecture, Wuhan City Polytechnic, Wuhan 430064, China

ABSTRACT

Social image platforms allow their users sharing and searching their photos based on images’ tags. These tags are provided by different users. Inevitably, the tags are spontaneously ambiguous, and personalized. So, learning the relevance between tags and images is playing an important role in tag-based retrieval systems. Choosing visual neighbors for seed images as voters is a widely used method for learning tag relevance. However, most existing methods of choosing visual neighbors for seed images are based on the global features of the whole images, ignoring the local features. In this paper we propose a pixel voting method to choose the visual neighbors for seed images. Experiment shows that this method is a more natural way to measure the similarity of images. Based the selected neighbors we learn the tag relevance, and the experiment on the MIR Flickr dataset shows that our algorithm is effective in tag de-noising and tag ranking.

INTRODUCTION

Nowadays, more and more social networking sites, like Flickr, Twitter and Facebook, provide platforms to build social relationships among people who share similar interests and background by sharing pictures in most cases. In order to improve the efficiency of image retrieval effect, social media platforms allow people to provide tags for their images. However, these user-created tags are spontaneously vague and subjective. According to [1, 2], there are only 10% images having their most relevant tags at the first place. Therefore, more and more meaningful and challenge work based tag relevance learning algorithms is attracting people’s attention, such as tag de-noising [3-5], tag recommendation [6-8] and tag ranking [9-11], which are studied to improve social images retrieval efficiency.

Many outstanding works focus on the social image tags relevance learning from multiple perspectives: [12] proposes a tag recommender system by using social images’ meta-data, such as images’ GPS information and photo taken time, etc.; [1,13-18] study the tags relevance by textual information, for example [1, 13, 17,
18] calculate the tags concurrence probability and analyze the semantical relevance between tags, [14] analyzes the ontological relationship for tags, [1,15] assume that the important tags’ positions are prior to those irrelevant ones;[19,20] argue that tags are content-related to the image, and after comparing to other 3 factors, [19] concludes that the image content is the key factor for tag relevance learning.

We confirm the opinion of [19-21] that the image content is the key to learn tag relevance. However, different from previous work, we measure the similarity of images by pixel voting instead of the whole image.

![Image](image.png)

(a)Seed Image, Tags: clouds, lake, sunset, tree, water

(b)Tags: sky, structure, tree, plant life

(c)Tags: clouds, lake, plant life, sky, water

(d) Tags: sea, sky, sunset, water

(e)Tags: animal, clouds, plant life, sky

Figure 1. Visual neighbors of the seed image. Images (b)–(e) are the visual neighbors of the seed image (a). The highlight part shows the most similar pixels in the neighbor image.

As shown in Figure 1, the social image usually contains more than one topic. And when two images have the same tags, it is possible that these two images have similar visual features in part of pixels in the image. In order to calculate the relevance scores between the seed image and its tags, the visual neighbors of the seed image are selected. The more similar the two images are, the more common tags they have. In Figure 1, the image (b~e) are the visual neighbors of the seed image (a). The gray image next to the original images include lots of spot whose brightness and location show the every pixel’s similarity of the original images and seed image.
In this paper, we first put forward the method of pixel voting to measure the similarity for images. And then, we calculate the relevance of tags and image using neighbor voting method. In the end we test the tags’ relevance in tag-noising and tag ranking. The rest of this paper is organized as follows. Section 2 reviews the related work about neighbor voting method. Section 3 presents details of pixel voting method. Section 4 discusses the experimental results. Finally, we conclude the paper in Section 5.

RELATED WORK

Given a social image \( x \), the core of tag relevance learning is to calculate the relevance score between tag \( t \) and the image \( x \), denoted as \( f(x, t) \), where \( t \in T \) is the potential tag of the image \( x \), and \( T \) is the set of all the tags.

Neighbor voting method [20] assumes that the more neighbors who contain the same tag, the higher the relevance score is, and the tag relevance score is defined as (1):

\[
f(x, t) = \sum_{k=1}^{N} \text{score}(x, x_k) \ast \delta(x_k, t)
\]

The note \( x_k \) is defined as one of the \( N \) visual neighbors of the seed image \( x \). These visual neighbors are selected from the images dataset. And the \( \delta(x_k, t) \) is a Kronecker delta. If \( t \) is a tag of \( x_k \), \( \delta(x_k, t) = 1 \); otherwise \( \delta(x_k, t) = 0 \).

\( \text{score}(x, x_k) \in [0, 1] \) describe the similarity between two images. The higher the score, the higher the voting right of \( x_k \). [14] used the all tags’ sematic similarity of two images to calculate the similarity between image \( x \) and \( x_k \), that is \( \text{score}(x, x_k) \), ignoring the images visual information. [19, 20] extract the visual features for the social images, and then calculate the distance between query images and seed images. The distance functions include \( \chi^2 \), Euclidean distance, KL divergence, and etc.

These methods have obvious defects:

**Local features are masked by global features:** Global features are good at showing the overall visual effects. While in many cases, histogram value is used to describe the global features which masks the local features’ information in most case.

**Images are only partially similar to each other:** Social images contain more than one element. If two images share same elements, they are probably similar to each other and have same tags. The overall images comparison cannot reflect the characteristics of the image which are partial similarity to each other.

Many literatures [22, 23] focus on local and global similarity problems. Our work measures the similarity of images in pixel level, and the experiments show that this method is state-of-the-art in tag relevance studies.
IMAGE SIMILARITY

Sub Image Similarity Score

[23, 24] define the similarity score of sub image \( p_t \) and seed image \( \Omega \) by mutual information, which is shown in (2), where \( P \) is the feature point located in the sub image \( p_t \). And score \( s(P) \) is defined as the similarity score of the feature point \( P \) and the seed image.

\[
s(p_t) = MI(\Omega, p_t) = \frac{\log(p(p_t|\Omega))}{p(p_t)} = \log \frac{\prod_{P \in p_t} p(P|\Omega)}{\prod_{P \in p_t} p(P)} = \sum_{P \in p_t} \log \frac{p(P|\Omega)}{p(P)}
\]

\[
= \sum_{P \in p_t} \frac{\log \frac{1}{p(\Omega) + \frac{p(P|\Omega)}{p(P|\Omega)} p(\Omega)}}{p(\Omega)} = \sum_{P \in p_t} s(P)
\]

(2)

After numbering all the feature points in an image, and denoting them as \( P_j \), \( j=1..M \), the (2) can be written as (3):

\[
s(p_t) = (\alpha_{t1}, \alpha_{t2}, \ldots, \alpha_{tM}) \ast \left( \begin{array}{c} s(P_1) \\ s(P_2) \\ \vdots \\ s(P_M) \end{array} \right) = \omega_t \ast S(P)
\]

(3)

The row vector \( \omega_t = (\alpha_{t1}, \alpha_{t2}, \ldots, \alpha_{tM}) \), where \( \alpha_{tj} \in \{0,1\} \). If \( P_j \in p_t \), then \( \alpha_{tj} = 1 \), otherwise \( \alpha_{tj} = 0 \). \( S(P) = (s(P_1), s(P_2), \ldots, s(P_M))^T \) denotes all the features’ similarity scores.

Pixel Similarity Score

The query image can be partitioned into \( K \) sub images. And for each pixel \( t \), there are \( K_t \) sub images contain it. The pixel similarity score of \( t \) is defined as the expectation of all the \( K_t \) sub images’ similarity score, as shown in (4).

\[
s(t) = E(s(p_t))
\]

\[
= \sum_{i=1}^{K_t} p(p^{(i)}_t)s(p^{(i)}_t) = \frac{1}{K_t} \sum_{i=1}^{K_t} s(p^{(i)}_t)
\]

\[
= \frac{1}{K_t} \sum_{i=1}^{K_t} \omega^{(i)}_t \ast S(P) = W_t \ast S(P)
\]

(4)

The row vector \( W_t \) is defined as (5):
\[ W_t = \sum_{i=1}^{K_t} \frac{1}{K_t} \omega_t^{(i)} = \sum_{i=1}^{K_t} \alpha_t^{(i)} + \sum_{i=1}^{K_t} \alpha_t^{(i)} + \ldots + \sum_{i=1}^{K_t} \alpha_t^{(i)} \] (5)

That is \( W_t \) denoting the probability of the feature points locate in the sub images of the pixel \( t \). So we can adopt any probability function listed in TABLE I to calculate \( W_t \), where \( x \) is the distance of the pixel and the feature point.

**TABLE I. SEVERAL PROBABILITY FUNCTIONS FOR \( W_t \).**

| Common PDF | Probability density function |
|------------|----------------------------|
| **Exponential Distribution** | \( \exppdf(x|\mu) = \begin{cases} \frac{1}{\mu} \exp^{-\frac{x}{\mu}}, & \text{and } x \geq 0 \\ 0, & \text{and } x < 0 \end{cases} \) |
| **Folded Normal Distribution[25]** | \( \foldpdf(x|\mu,\sigma) = \begin{cases} \frac{1}{\sigma \sqrt{2\pi}} \exp \frac{(x-\mu)^2}{2\sigma^2} + \frac{1}{\sigma \sqrt{2\pi}} \exp \frac{(x+\mu)^2}{2\sigma^2}, & x \geq 0 \\ 0, & x < 0 \end{cases} \) |
| **Reciprocal Distribution[26]** | \( \recippdf(x,a,d_{max}) = \begin{cases} \frac{1}{a x + 1}, & \text{and } x \in (0,d_{max}) \\ 0, & x \notin (0,d_{max}) \end{cases} \) |

**Pixel Voting**

Given the query image \( x \), there are \( m \times n \) pixels and \( M \) feature points in it. Each feature point can affect all the pixels’ similarity score. After calculating the distance between each pixel and the feature point, the similarity score maps can be obtained. Each similarity score map can be designed as a probability map which is calculated on a selected probability density function as list in TABLE I.

![Figure 2. Similarity score map.](image)

This process can be calculated in parallel for all the \( M \) feature points, as shown in the Figure 2. Many kinds of features can be used here, such as SIFT[27], SURF[28], FAST feature[29] etc. These kinds of feature include interest points’ information and their locations. The feature points of an image are denoted as red triangles and blue points in the Figure 2. Compared to the average feature value of all the features in the dataset, red triangles are more similar to the seed image, and
the blue points are not. All the Maps overlap each other and form the whole similarity score map.

Each pixel can vote to determine the similarity score of query image and seed image by (6), \( \alpha \) is the threshold. If the similarity score of pixel \( t \) is greater than \( \alpha \), the two value function \( \delta(s(t) \geq \alpha) = 1 \), otherwise \( \delta(s(t) \geq \alpha) = 0 \).

\[
\text{score}(x, x_s) = \frac{1}{m \cdot n} \sum_{t \in x_s} \delta(s(t) \geq \alpha)
\]

\[ (6) \]

EXPERIMENT

The experiment data we used is composed of 3000 social images from MIR Flickr dataset [30]. Each image has a potential tag list and a ground truth tag list. We test our algorithm with 14 common tags, including animal, baby, bird, car, clouds, dog, flower, food, lake, night, people, sky, sunset, tree, and etc.

SURF descriptor is a scale- and rotation-invariant descriptor for an image features. It can be computed and matched much faster compared to SIFT feature. And this kind of descriptor includes two parts, one part is feature information, and the other part is the location of interest point. Bag-of-Words method ignores the descriptors’ location part, and change the descriptors feature information into scalar data which helps to simplify the calculations.

After extracting SURF feature for all the images, bag of words method is used to quantize the features, and the size of the vocabulary is 5000. In (2), the parameters are set as: \( p(\Omega) = p(\Omega^c) = 0.5 \), and distance function is \( p(P|\Omega) = \text{NHI}(f_P, f_\Omega) = \frac{|f_P \cap f_\Omega|}{|f_P \cup f_\Omega|} \in [0,1] \), threshold \( \alpha \) in (6) is set as 0.

Parameters of Several Probability Functions

In this section, we calculate the parameters of the three probability density functions listed in TABLE I. After analysis, those probability density functions need to meet the following three constraints: (1) For the pixel located the same position as the feature point, its probability is defined as \( p_{\text{max}} = 1 \); (2) For the farthest pixel away from the feature point, its probability is defined as \( p_{\text{max}} = \epsilon > 0 \), where \( \epsilon \) is a very small number; (3) All the Cumulate Distribution Function is less than 1, which also can be written as: \( \int_{0}^{d_{\text{max}}} pdf \ dx = 1 - \epsilon' \), where \( d_{\text{max}} \) is the max distance of pixels and feature point, pdf is the probability density function, and \( \epsilon' \) is another a very small number.

And then we can calculate the parameters in all the probability density functions:
a) **Exponential Distribution**

According to the constraint (1), \( \mu = 1 \), the calculation process is shown in (7):

\[
p_{max} = \text{exppdf}(x|\mu)|_{x=0} = \frac{1}{\mu} = 1
\]  

(7)

According to the constraint (2), we can normalize the distance of pixels and feature point as \( d \in [0, d_{max}] \), \( d_{max} = -ln(\varepsilon) \), which calculation process is shown in (8):

\[
p_{min} = \text{exppdf}(x|\mu)|_{x=d_{max}} = \exp(-d_{max}) = \varepsilon
\]  

(8)

b) **Folded Normal Distribution**

The folded normal distribution is a probability distribution related to the normal distribution. Given a normally distributed random variable \( x \) with mean \( \mu \) and variance \( \sigma^2 \), the random variable \( y = |x| \) has a folded normal distribution.

For \( p_{max} = \text{findpdf}(x|\mu, \sigma)|_{x=\mu} \), and according to constraint 1, we can set \( \mu = 0 \); And according to PaTa criterion \( \int_0^{3\sigma} pdf \ dx = 0.997 \), then we can set \( d_{max} = 3\sigma \). So we can normalize the distance of pixels and feature point as \( d \in [0, d_{max}] \), where \( d_{max} = 1 \), and \( \sigma = 1/3 \).

c) **Reciprocal Distribution**

According to the constraint (3), the Cumulate Distribution Function is shown as (9):

\[
F(d_{max}) = \int_0^{d_{max}} \frac{1}{ax + 1} \ dx = 1
\]

\[
\ln \left( \frac{1}{ax + 1} \right)|_0^{d_{max}} = a
\]

\[
d_{max} = \frac{\exp(a)-1}{a}
\]

(9)

So if we normalize the distance of pixels and feature point as \( d \in [0, d_{max}] \), where \( d_{max} = 1 \). According to constraint (2), \( p_{min} = \frac{1}{ax+1}|_{x=d_{max}} = \frac{1}{a*d_{max}+1} = \exp(a) \), if we set \( a = 5 \), then \( p_{min} = 0.0067 \) is a very small number.

**Similar Image List**

In this experiment, we adopt exponential probability density function as the function of \( W_t \) where \( \varepsilon \) is set to 0.001. After extracting global feature, neighbor voting algorithm is adopted as the baseline comparing to the pixel voting algorithm.
in selecting the top-9 similar images as shown in Figure 3. The left column is the seed image, and the right column is the top-9 similar images list according to their similar scores: the line (a) is the result of pixel voting method, and the line (b) is the result of the neighbor voting. The irrelevant image is shown in the red box.

Tag De-noising

The goal of tag de-noising is to remove irrelevant tags of social images. In this section, six algorithms are used for contrast experiments. Three of them are baseline methods including neighbor voting [20], weighted neighbor voting [3] and rank based neighbor voting [19]. And another three Pixel Voting algorithm are based different distributes.

The MIR Flickr dataset used here includes a potential tag list and a ground truth tag list for each image. Eliminating the irrelevant tags from the potential tags for each image is the goal of tag de-noising. Take the ‘baby’ tag for example, the seed image’s visual neighbors are selected according to (6). And the visual neighbors vote to calculate the relevance score of the tag ‘baby’ and the seed image. Then all the images are sorted inversed. The ground truth tag lists are used to class the images into positive and negative. According to the prediction result is correct or not, the result are divided into true positive (TP), false positive (FP), true negative (TN) and false negative. Then the TPR and FPR are calculated according to (10) and (11). Then the ROC curves can be plotted.

\[
TPR = \frac{TP}{TP + FN} \tag{10}
\]
\[
FPR = \frac{FP}{FP + TN} \tag{11}
\]

Three tags ROC curves are shown in Figure 4. The baseline is shown in dashed lines, and the Pixel Voting algorithm are shown in solid lines.

Another common measurement of evaluating the performance is Average Precision shown as (12), where \( R \) is the number of all the relevant images in image set, \( R_j \) is the number of relevant images among the top \( j \) ranked images, and if the \( j \)-th images is relevant \( I_j = 1 \), otherwise \( I_j = 0 \). And the result is shown in TABLE II.

\[
AP = \frac{1}{R} \sum_{j=1}^{n} \frac{R_j}{j} I_j \tag{12}
\]

The result shows pixel voting is good at fine category tag de-noising like baby, sunset, and etc. But the performance of pixel voting is not good at the coarse categories tags like animal and food, in which categories images are greatly different in details.
**Tag Ranking**

The tag’s position is an important factor in tag-base image searching. In this experiment, each image’s tags are classified into 5 levels: perfect relevant, excellent relevant, good relevant, fair relevant and bad relevant [1]. And we use NDCG score to evaluate the performance of tag ranking compared to the baseline methods.

| Seed Image | Sorted Similar Images |
|------------|-----------------------|
| baby       | (a) ![baby](image1)    |
|            | (b) ![baby](image2)    |
| bird       | (a) ![bird](image3)    |
|            | (b) ![bird](image4)    |
| dog        | (a) ![dog](image5)     |
|            | (b) ![dog](image6)     |
| sunset     | (a) ![sunset](image7)   |
|            | (b) ![sunset](image8)   |

Figure 3. Top 9 similar image list of the seed image.
Normalized Discounted Cumulative Gains (NDCG)[18] is a widely used measurement of tag ranking. The NDCG score is computed as (11), where $r(i)$ is the relevance level of the $i$-th tag, $K$ indicates that NDCG scores are calculated using the top $K$ ranked tags, and $z$ is a normalization constant ensuring the cumulative gain of NDCG score is 1. In this experiment we set $K$ to 5, 7, and 9. The result is shown in the Figure 5. It shows our method gains the highest NDCG scores compared to the other methods.

$$NDCG@K = z \sum_{i=1}^{K} \frac{2^{r(i)}-1}{\log(1+i)}$$

Figure 4. The ROC curves of three tags.
CONCLUSIONS

Since user-created tags are noisy and personalized, the relevance of the tags and images is important in the tag-based retrieving including tag de-noising and tag ranking. Experiment shows that visual neighbors of seed images have the best voting rights to determine the relevant tags for each social image. Most social images contain more than one visual element and there are only part of those elements is similar to the other social images. In order to measure the detailed similarity for images, we propose the pixel voting method to choose the visual neighbors for seed images, and these visual neighbors have weighted voting right based their similarity score. Experiment shows that the method is a more natural way to measure the similarity of images, and it is effective in tag de-noising and tag ranking.

Because of the semantic gap between the images and the tags, tag relevance learning cannot be completely resolved. For example, the tortoise image and the giraffe image have the same tag animal, but there are greatly different in images visually. So in the future, we have to learn the tag relevance from cross media semantic understanding to build shared semantic space.

REFERENCES

1. Liu, D., Hua, X. S., Yang, L., Wang, M., and Zhang, H. J. 2009. “Tag ranking,” International Conference on World Wide Web, pp.351-360. ACM.
2. He, Y., Kang, C., Wang, J., Xiang, S., and Pan, C. 2015. “Image tag-ranking via pairwise supervision based semi-supervised model,” Neurocomputing, 167(C), pp. 614-624.
3. Lee, S., Neve, W., and Ro, Y. M. 2014. “Visually weighted neighbor voting for image tag relevance learning,” Multimedia Tools and Applications, 72(2), pp. 1363-1386.
4. Tian, F., Shen, X. K., Shang, F. H., and Zhou, K. 2012. “Automatic Image Tagging by Multiple Feature Tag Relevance Learning.” *Pattern Recognition*. Springer Berlin Heidelberg.

5. Li, X., Snoek, C. G. M., and Worring, M. 2010. “Unsupervised multi-feature tag relevance learning for social image retrieval,” *ACM International Conference on Image and Video Retrieval*, 324, pp.10-17. ACM.

6. Font, F., Serrà, J., and Serra, X. 2015. “Analysis of the impact of a tag recommendation system in a real-world folksonomy,” *ACM Transactions on Intelligent Systems and Technology*, 7(1), pp. 1-27.

7. Sun, A., Bhowmick, S. S., and Chong, J. A. 2011. “Social image tag recommendation by concept matching,” *ACM International Conference on Multimedia*, 14, pp.1181-1184. ACM.

8. Mcparlane, P. J., Moshfeghi, Y., and Jose, J. M. 2013. “On contextual photo tag recommendation,” pp. 965-968. ACM.

9. Feng, S., Lang, C., and Li, B. 2012. “Towards relevance and saliency ranking of image tags,” *ACM International Conference on Multimedia*, 110, pp.917-920. ACM.

10. Zhuang, J., and Hoi, S. C. H. 2011. “A two-view learning approach for image tag ranking,” *ACM International Conference on Web Search and Data Mining*, pp. 625-634. ACM.

11. Li, M., Tang, J., Li, H., and Zhao, C. 2012. “Tag ranking by propagating relevance over tag and image graphs,” *International Conference on Internet Multimedia Computing and Service*, pp.153-156. ACM.

12. Xuming Qian, Xiaoxiao Liu, Chao Zheng, Youtian Du, and Xingsong Hou. 2013. “Tagging photos using users’ vocabularies,” *Neurocomputing*, 111(111), pp. 144-153.

13. Sun, F., Li, H., Zhao, Y., Wang, X., and Wang, D. 2013. “Towards tags ranking for social images”. *Neurocomputing*, 120(10), pp. 434-440.

14. Zhu, S., Ngo, C. W., and Jiang, Y. G. 2012. “Sampling and ontologically pooling web images for visual concept learning,” *IEEE Transactions on Multimedia*, 14(4), pp. 1068-1078.

15. Zhang, J., Liu, X., Zhuo, L., and Wang, C. 2015. “Social images tag ranking based on visual words in compressed domain,” *Neurocomputing*, 153, pp. 278-285.

16. Chaoran Cui, Peiguang Lin, Xiushan Nie, Yilong Yin, and Qingfeng Zhu. 2017. “Hybrid textual-visual relevance learning for content-based image retrieval,” *Journal of Visual Communication and Image Representation*.

17. Chen, J., Liu, Y., Yang, G., and Zou, M. 2017. “Inferring tag co-occurrence relationship across heterogeneous social networks,” *Applied Soft Computing*.

18. Uricchio, T., Ballan, L., Seidenari, L., and Bimbo, A. D. 2016. “Automatic image annotation via label transfer in the semantic space,” *Pattern Recognition*.

19. Ba, Q. T., Sun, A., and Bhowmick, S. S. 2012. “Content is still king: the effect of neighbor voting schemes on tag relevance for social image retrieval,” *ACM International Conference on Multimedia Retrieval*, pp. 9. ACM.

20. Li, X., Snoek, C. G. M., and Worring, M. 2009. “Learning social tag relevance by neighbor voting,” *IEEE Transactions on Multimedia*, 11(7), pp. 1310-1322.

21. Xia, Z., Feng, X., Lin, J., and Hadid, A. 2017. “Deep convolutional hashing using pairwise multi-label supervision for large-scale visual search,” *Signal Processing Image Communication*. 
22. Dawood, H., Dawood, H., and Guo, P. 2013. “Global matching to enhance the strength of local intensity order pattern feature descriptor,” *International Conference on Advances in Neural Networks*, 7951, pp.497-504, Springer-Verlag.

23. Meng, J., Yuan, J., Jiang, Y., Narasimhan, N., Vasudevan, V., and Wu, Y. 2010. “Interactive visual object search through mutual information maximization,” *International Conference on Multimedia*, 2010, Firenze, Italy, October, pp. 1147-1150. DBLP.

24. Jiang, Y., Meng, J., Yuan, J., and Luo, J. 2015. “Randomized spatial context for object search,” *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 24(6), 1748.

25. Tsagris, M., Beneki, C., and Hassani, H. 2014. “On the folded normal distribution,” *Mathematics*, 2(1), pp. 12-28.

26. Hamming, R. W. 1970. “On the distribution of numbers,” *Bell System Technical Journal*, 49(8), pp. 1609-1625.

27. Ke, Y., and Sukthankar, R. 2004. “PCA-SIFT: a more distinctive representation for local image descriptors,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, pp.506-513, IEEE Computer Society.

28. Bay, H., Tuytelaars, T., and Gool, L. V. 2006. “Surf: speeded up robust features,” *Computer Vision and Image Understanding*, 110(3), pp. 404-417.

29. Dollar, P., Belongie, S., Belongie, S., and Perona, P. 2014. “Fast feature pyramids for object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8), pp. 1532-1545.

30. http://press.liacs.nl/mirflickr/