Insight Centre for Data Analytics (DCU) at TRECVid 2014: Instance Search and Semantic Indexing Tasks

Kevin McGuinness[1*], Eva Mohedano[1], ZhenXing Zhang[1], Feiyan Hu[1],
Rami Albatal[1], Cathal Gurrin[1], Noel E. O'Connor[1], Alan F. Smeaton[1],
Amaia Salvador[2], Xavier Giró-i-Nieto[2], Carles Ventura[2]

[1]Insight Centre for Data Analytics, Dublin City University, Ireland
[2]Image Processing Group, Universitat Politècnica de Catalunya, Spain

Abstract

Insight-DCU participated in the instance search (INS) and semantic indexing (SIN) tasks in 2014. Two very different approaches were submitted for instance search, one based on features extracted using pre-trained deep convolutional neural networks (CNNs), and another based on local SIFT features, large vocabulary visual bag-of-words aggregation, inverted index-based lookup, and geometric verification on the top-N retrieved results. Two interactive runs and two automatic runs were submitted, the best interactive runs achieved a mAP of 0.135 and the best automatic 0.12. Our semantic indexing runs were based also on using convolutional neural network features, and on Support Vector Machine classifiers with linear and RBF kernels. One run was submitted to the main task, two to the no annotation task, and one to the progress task. Data for the no-annotation task was gathered from Google Images and ImageNet. The main task run has achieved a mAP of 0.086, the best no-annotation runs had a close performance to the main run by achieving a mAP of 0.080, while the progress run had 0.043.

1 Introduction

In the context of our continuous participation in TRECVid [16], the DCU-Insight team this year comprises researchers and developers from Insight Centre for Data Analytics – Dublin City University (Ireland), and from the Image Processing Group – Universitat Politècnica de Catalunya (Spain). In the 2014 TRECVid benchmarking [12], the team participated in two tasks, Semantic Indexing (SIN) and Instance Search (INS). Motivated by our previous participation in various TRECVid tasks, this year we decided to evaluate advanced techniques in classification and features extraction for effective and effective and efficient recognition and visual matching.

The two participating teams deal with a wide range of research related to Video Indexing and Retrieval, Visual Lifelogging semantic extraction, Machine Learning and Image processing. Hence the focus this year was on the Instance Search and the Semantic Indexing tasks. Our experiments consisted of four runs submitted in the INS task and another four in the SIN task. One of main focus was the evaluation of the performance of pre-trained convolutional neural networks (CNN), the goal is to explore the possibility of using such tools for high level features extraction [17] and for visual matching and retrieval.

This paper is structured as follow: in Section 2, we describe our approaches in the Instance Search task. Section 3 focuses on the Semantic Indexing task, then Section 4 summarizes and concludes this paper.

[*] Contact author: kevin.mcguinness@insight-centre.org

2 Instance Search

This submission was carried out by two independent groups in the Insight-DCU team. The teams will be referred as *INS-1* and *INS-2* during the rest of this paper. The first one used pre-trained convolutional neural networks to produce the image features and was responsible of I_D_insightdcu_1, I_D_insightdcu_2, F_D_insightdcu_3 runs. The second used an extension of previous work by Insight-DCU [19] based on bags-of-visual-words of SIFT descriptors, and was submitted as the F_D_insightdcu_1 run. The following sub-sections describe the runs in detail.

2.1 INS-1 approach

Our first approach was based on using pre-trained convolutional neural networks (CNN) to extract image descriptors. Features from these networks have been shown in recent works [13, 15] to give strong results in a variety of classification and retrieval tasks. Two of the submitted runs also used user interaction to manually annotate the retrieved shots thereby helping to improve the performance of the system. In this section, a detailed description of the different parts of the system is presented.

2.1.1 Target Dataset

A target image dataset was built by uniformly extracting keyframes for every shot with a sample rate of 1/4 fps. The resulting dataset contained 647,628 keyframes and had a size of 66GB (referred to in the following sections as the 'Full Dataset'). In addition, a subset was also used during the development stage of the retrieval system. This subset consists of only the relevant shots for each query topic of TRECVid Instance Search 2013. This subset consisted of 23,614 keyframes, which was used to quickly test the different approaches during the development of the pipeline. This dataset will be referred to as the 'Ground Truth Subset' in the following sections.

2.1.2 Feature-Extraction

The system used *Caffe* [7], a publicly available code, to extract CNN features of the images. *Caffe* also provides pre-trained ImageNet models [14] that can be used off-the-shelf to directly obtain feature vectors. Convolutional neural networks are composed of different layers, each one encoding different parts and features of the image. While lower levels focus on details and local parts, upper levels contain a more global description of the image. For this reason, it has been suggested in several works [10, 3] that the features from the upper layers of the CNN are the best ones to be used as descriptors for image retrieval. Following those insights, our system uses Layer 7 of the network as global feature vectors.

Global CNN features are useful to describe the general appearance of the image, but the goal of INS is not just to retrieve similar images, but images that contain a specific object. Because TRECVid provides the binary masks of the objects in the topic images, our first intuition was to compute the CNN features using as input a cropped version of the image, keeping only the pixel values inside the binary mask. This way, we obtain a representation of the object itself, not the whole image. However, these new local features on the query images should be matched with local features on the target database, for which no binary masks are provided. To make this possible, object candidates were computed using the SCG algorithm from [4] for all the keyframes in the target database. This way, local features can be computed in both the query and target sets.

2.1.3 Feature matching

The feature matching of the query and keyframe feature vectors for the experiments with the Ground Truth Subset was performed exhaustively, by computing their cosine similarities to the query vector and sorting them in descending order. However, this approach was not feasible with the Full Dataset, for which the number of keyframes in the target database is much higher. In this case, to quickly retrieve the most similar keyframes for a query topic, feature vectors are embedded in a Hamming space building up a forest of 50 binary trees by using random projections. Similar features are then retrieved using approximate nearest neighbor (ANN) indexes. We use the Spotify Annoy implementation for ANN indexing and lookup.

2.1.4 Ranking the results

The keyframes (either all of them for the Ground Truth Subset or the ones retrieved using ANN for the Full Dataset) are scored according to the cosine similarity between their feature vectors and the query feature vector, and sorted in descending order to produce a ranked list. However, this procedure is not sufficient to generate the final ranking. As mentioned earlier, TRECVid provides 4 images for each query topic. Using global CNN features, each query topic would consist of 4 feature vectors that are used separately to retrieve similar images from the target database, consequently producing 4 independent rankings, which need to be merged in a single one. This is done by fusing them and sorting them again by their score. Then, if a keyframe appears several times, the score associated to it would be the maximum among all (max-pooling). The final step to produce the final ranking is to group the keyframes that belong to the same shot, in order for them to represent a single entry of the ranking. Again, the keyframe with maximum score is kept as the one representative of the shot (max-pooling).

2.1.5 User Interface

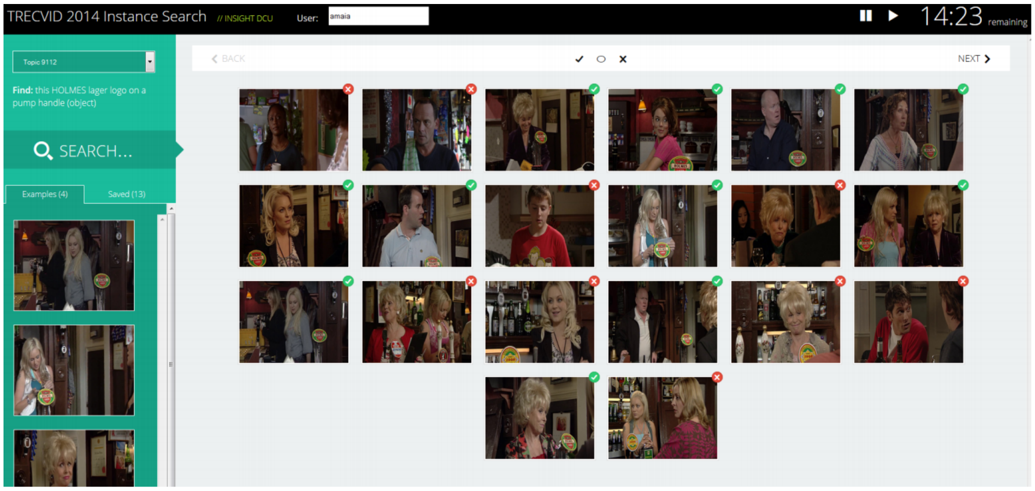
This section introduces the interface used to collect annotations for our submission to TRECVid Instance Search 2014. Figure 1 shows a screenshot of the interface. Its different parts and possible user interactions are:

- A dropdown list containing the IDs of the query topics (top-left). Users can select the one that they want to use to retrieve similar results.
- A textual description of the query topic along with its category (object, location or person).
- An expandable tag Examples, displaying 4 images containing the query topic (left-column).
- An expandable tag Saved, displaying the positive images annotated by the user (left-column).
- A textbox to introduce the user name (top-center bar).
- The results panel. It displays the retrieved keyframes by the system to the user. The user's task is to annotate them as either positive or negative.
- The Search button. Users can press it at the beginning and after they have annotated all the keyframes they see in the results panel to obtain different ones.
- A counter displaying the amount of time remaining for a query topic. It is possible to pause and unpause it by the user. This interface was built using HTML5 and AngularJS on the client side and Python on the server side, using MongoDB as connected database.

2.1.6 Automatic run: F_D_insightdcu_2

Run description This submission was fully automatic, and it obtained a mAP of 0.062. It consists in two steps:

1. Generating an initial ranking of 1000 shots based on global CNN descriptors.
2. Re-ranking the retrieved shot lists obtained by global CNN features using the concatenation of local and global CNN descriptors.

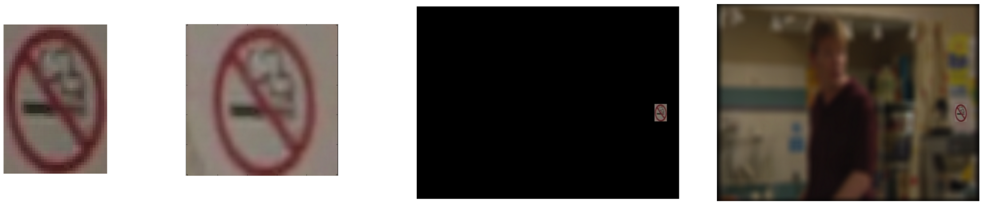


User: amaia

Topic 9112

Find: this HOLMES lager logo on a pump handle (object)

Figure 1: Screenshot of the interface for TRECVid Instance Search 2014.



(a) Cropping (b) Square crop (c) Zeroing (d) Blurring

Figure 2: Different local approaches:(a) *Cropping* based on the bounding box that contained the object of interest. (b) *Square crop* containing the object of interest. (c) *Zeroing* the background. (d) *Blurring* the background.

Experiments that defined the run configuration All the images or local image paths were re-sized to 227×227 pixel resolution to fit with the neural network input and extracting that way their CNN feature (section 2.1.2). In order to add local information to the image descriptors, a set of object candidates per image frames were computed by using [4]. Consequently, there were several variables that needed to be set for this run. Mainly we had to define: (1) What kind of masking or cropping strategy should we use in order to compute the CNN descriptors at local level. (2) How many object candidates should we consider per image frame. (3) How to combine the global and local features within an image frame.

(1) Setting the cropping strategy for the local paths. We considered 4 different approaches for cropping and masking the image frames and extracting the CNN features by using local information (Figure 2). In order to evaluate and compare the different approaches we used as a toy subset. This subset contained 120 query images of TRECVid 2013 where each query frame had only 3 relevant images.

Comparing the different approaches in this small subset, we found that the best strategy was the square crop of the local path. Additional experiments revealed that adding some pixels of context helped in the retrieval performance. Then, the local regions were cropped by fixing a square region and the, incrementing their square sizes by adding the 25% of the original square size to each size (named square crop with 1/4

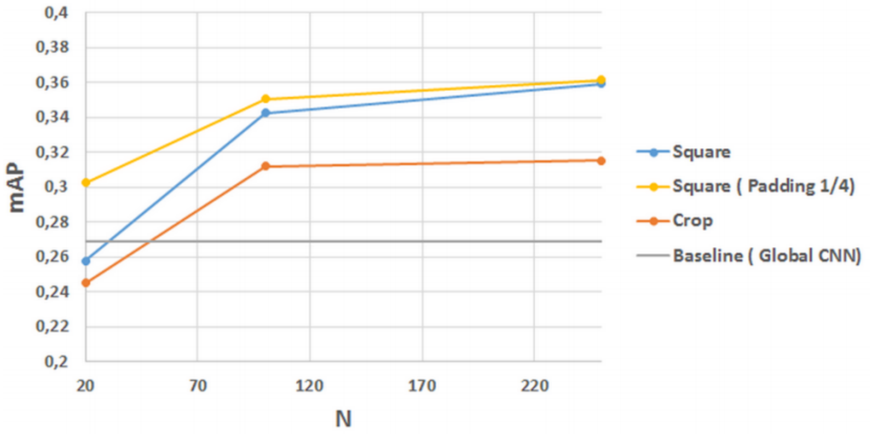


Figure 3: mAP considering different local approaches and number of candidates (N).

padding).

(2) Setting the number of object candidates. A different number of object candidates per frame was extracted for the same toy subset. Now, each frame consisted in a set of N treated independently. The final ranking was performed as described in Section 2.1.4. Despite the results (Figure 3) show that the higher the number of candidates the better the systemś performance is, the number of candidates had to be restricted to 20 due to time constrains and lack of computational power when dealing with the Full dataset.

(3) Combining local and global information. Once the a decision was made to compute the local CNN descriptors, the performance was evaluated on the Ground Truth subset. The first local approach consisted in replacing the global features by the local ones using squared crops with context information (1/4 padding). Then, we tried two approaches combining global and local features:

- Concatenation: CNN features for both global and local approaches are concatenated in a single feature vector.
- Aggregation: CNN features for both global and local approaches are merged but treated independently (global and local feature vectors for the query images are compared to both global and local feature vectors of target dataset).

Figure 4 compares the performance of the system using global CNN features with the different local approaches mentioned above. The results indicate and increase of performance only when using the concatenation of global and local CNN features. However, due to the computational challenges and memory issues that we were facing with the Full dataset, we could only use the local approach as a re-ranking technique after obtaining a ranking using global features. By doing so, mAP increases from 0.1467 to 0.1663. This new ranked lists were the ones displayed in the user interface at the beginning (i.e. the first time that a user performs a search for a topic).

Summary and comments The different experiments on the query and Ground Truth subsets were the based of the run that we submitted. This run, as described in the beginning of this section, consisted then in generating the automatic run based on global CNN features. After that, and due to resources limitations, we only extracted the first 20 object candidates per keyframe within the first 1000 relevant shots of the ranking. We cut into squares the candidates adding a bit of context and we computed the CNN on them.

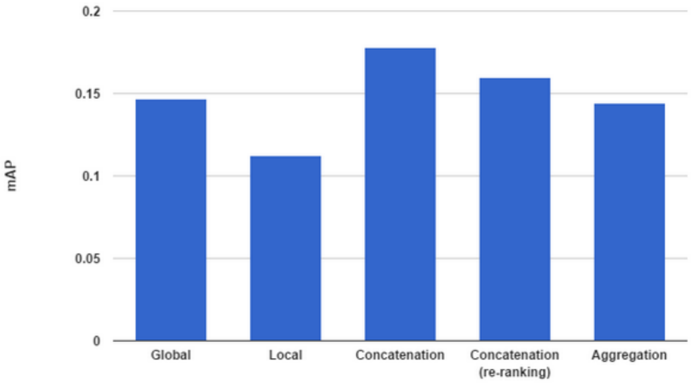


Figure 4: Results using local features.

Local and global CNN descriptors were combined by concatenation. The final result of 0.062 mAP, points out the difference in performance when comparing the full and ground truth subset.

Even though the result was not good, we used the automatic list in combination with the user annotations and relevance feedback techniques to improve the system performance. (Section 2.1.7)

2.1.7 Interactive runs: I_D_insightdcu_1 and I_D_insightdcu_2

Run descriptions Four runs were submitted with the interface introduced in Section 2.1.5. Users were asked to annotate the results of the ranking list as either relevant (positive) or non-relevant (negative). This information can be used in several ways in order to improve the ranking. We implemented two approaches of relevance feedback, corresponding to the two interactive runs presented:

- **Query expansion** (I_D_insightdcu_1, 0.135 mAP). One thing that can be done is to use the positively annotated keyframes as query images. This means that the ranking produced by each one of those keyframes will be merged with the ones produced by the topic images as explained in Section 2.1.4. Conversely, negative annotations are removed from the ranking.
- **SVM Scoring** (I_D_insightdcu_2, 0.126 mAP). Another relevance feedback strategy is to train a classifier using positive and negative annotations collected with the user interface. Then, this classifier can return confidence scores for each one of the keyframes in the database. Then, the new ranking would contain the keyframes sorted by the score returned by the classifier. In this case, both a Linear SVM and a SVM with RBF kernel were tested.

For both configurations, the new ranking was built by (1) pushing the positive annotations to the top of the ranking, (2) adding the results of the relevance feedback strategy and (3) removing the negative annotations.

Experiments with simulated annotations To be able to estimate the impact of these relevance feedback techniques before the submission, the Ground Truth subset was used. The adopted strategy consisted in simulating the userś annotations and using them to evaluate the different relevance feedback approaches. To do so, the results using local and global features for re-ranking were taken as a baseline list. The positive and negative annotations were simulated by taking all the relevant and non relevant keyframes of the ranking, respectively. This experiment can be repeated several times changing the percentage of the ranking that is being observed, which could simulate the relation between the amount of effort done by the user and the mAP.

Figure 5 shows the results of this experiment by simulating different levels of user effort by truncating the ranking at different positions. As expected, the higher number of annotations, the higher the mAP. It can also be observed that the Linear SVM strategy greatly outperforms the one using query expansion.

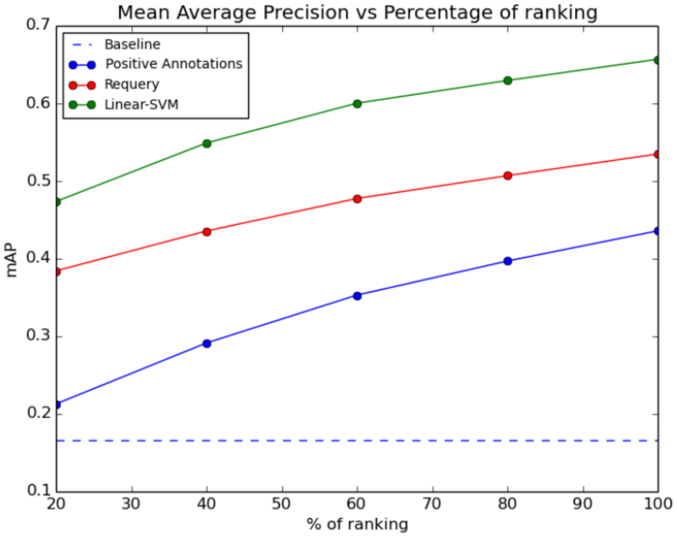


Figure 5: Simulation of mAP VS Percentage of ranking observed by the user using different relevance feedback strategies. Baseline is the mAP of the original ranking. Positive Annotations displays the result of a new ranking built only with the positive annotations contained in the observed ranking. Finally, Re-query shows the results for the query expansion technique and Linear-SVM gives the results for the SVM scoring technique (both explained earlier in this section.).

Summary and comments Although the CNN descriptors do not work too well for the retrieval task, we could see from the simulations that by adding relevance feedback techniques to the system it was possible to highly increase the mAP of the fully automatic approach. In the submitted interactive runs, we could increase from the automatic 0.062 mAP to 0.126 and 0.135 mAP (Query expansion and SVM scoring, respectively). This increase in performance highlight the potential of this techniques, event using simple approaches for that. Future work will explore to improve the way to use these user's annotation and train more sophisticated models to generate the scoring for the re-ranking.

2.2 INS-2 approach

The INS-2 team developed an instance search system based on the work from last year's participation [19]. This scalable visual object retrieval approach uses the widely-used bag-of-word representation and vector-space model of information retrieval. Learning from the experience of last year, we used high-dimensional vocabulary size to increase the visual words' discriminative power, and then spatial verification and query expansion technologies to further improve the performance. Specifically in this year's experiment, we focused on increasing the initial ranked accuracy by using a query-adaptive weighting function for visual objects similarity measurement which is a crucial point for spatial verification and query expansion. Product quantization for Approximate Nearest Neighbor Search is also used to improve the accuracy and speed performance in the feature quantization step.

Experiment Configuration This paragraph describes the implementation configuration for our visual object retrieval system. After extracted the dense frames for each video in the test dataset, we used the ColorDescriptor library [18] tool to detect the affine invariant interest regions (Harris-Laplace) and then described them with SIFT descriptors. To reduce quantization error and increase the discriminative power for

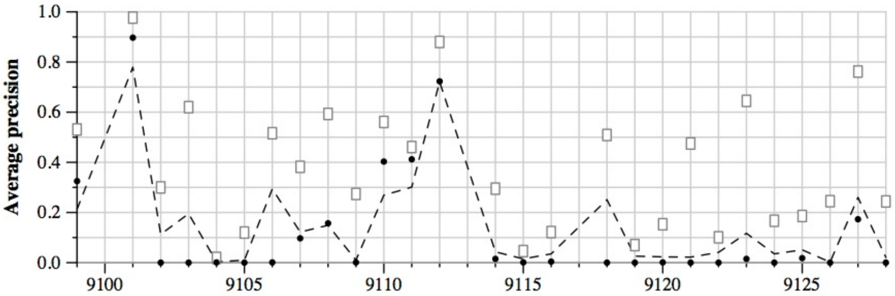


Figure 6: AP results overview all INS topics (Run score (dot) versus median (—) versus best (box) by topic)

every visual word, we trained a vocabularies of one-million clusters. Then all the descriptors for each frame were assigned to the closest cluster to generate a sparse feature vector to represent the frame image. For efficiency, the retrieval framework stored the visual word in an inverted index structure. Up to this point, the visual object retrieval system was completed, and we used all the four query images to perform the retrieval task. A simple yet well performing multiply query method, jointed average query strategy [2], was used to create an average BoW vector to query the database. After we got the initial results, spatial verification was applied to the top-100 ranking results to verify the consistency of the retrieval images with the query regions. Finally the we picked the top-20 results to expend the query which also used average query strategy.

The *TF-IDF* weighting scheme was used in our last year's experiment. The basic idea is to assign more weight to therms with less frequency between documents and high occurrences inside document. As mentioned in [8], this traditional weighting scheme does not address the problem of visual word *burstiness*. Burstiness means a given visual element is more likely to repeatedly occurred in one image due to the fact of similar background, scene or texture. It is normally happened in visual content and harmful for the object retrieval accuracy. So this year we adopted the L_p-*Norm IDF* [20] weighting scheme. According to our understanding, this weighting scheme consider the relation between visual words and rank the images which had more connections with query topic images firstly.

Results and Discussion For this approach we submitted one fully automatic run for evaluation, due to the fact that we also use the CNN features and submitted three runs (one fully automatic run and two interactive runs). We achieved a mAP of 0.120 and our best result among all topics achieved a mAP of 0.897. As shown in Figure 6, our performance followed the same trends as the median results which mainly because most of the system use the same approach as we did. In the category level, "PERSON" category got a lack performance, all five topics (9104, 9115, 9116, 9119, 9124) has Median AP near to 0, which indicated that this approach is not competitive for Human retrieval in large video collections. And then In "OBJECT" category, Logos of product or sign (9106, 9112, 9118, 9109, 9126) had inconsistent performance. After looking at the search images, a simple guess would be small object with rich and unique texture can also achieve great success in this approach, otherwise the performance would be very limited. This two observation draw us much attentions and would be our further research directions.

3 Semantic Indexing

Insight-DCU has submitted four runs for TRECVid 2014 semantic indexing task.

- **Main task** 2B_M_A_insightdcu.13_1
- **No annotation task** 2B_M_E_insightdcu.14_1 (RBF) and 2B_M_E_insightdcu.14_2 (Linear)
- **progress task** 2B_M_A_insightdcu.14_1

3.1 Datasets

The development dataset used for main and progress tasks combines the development and test datasets of the 2010 and 2011 SIN tasks: IACC.1.tv10.training, IACC.1.A, IACC.1.B, and IACC.1.C.

The no annotation task aims to automatically gather training examples using available resources on the Internet. Several recent papers have demonstrated the effectiveness of such an approach. [6] used search engine results to gather material for learning the appearance of categories, [5] shows that effective classifiers can be trained on-the-fly at query time using examples collected from Google Image search. The AXES research search engine [11] uses a combination of pre-trained classifiers and on-the-fly classifiers trained using examples from Google Image search. [9] investigate four practices for collecting training negative and positive examples from socially tagged videos and images.[1] has described a general framework combining natural language processing to more precisely retrieve training samples from the Internet.

For this no annotation task single-term queries were posted to two data sources: Google Images, and ImageNet (an image database organized according to the nouns of the WordNet hierarchy where each node is depicted by an average of +500 images). Unlike results from Google Images, examples gathered from ImageNet are classified by human annotators, and are therefore a “purer” source of training images. To ensure a high-quality training set, we first search for the concept in ImageNet; if the concept does not exist in as an ImageNet visual category, we use images retrieved using a search for the term on Google Images. We gathered 36 concepts from ImageNet and 24 concepts from Google Image search. The mean number of samples of each concept collected from ImageNet is 1747 (Std. 972), mean while the average number of samples of each concept gathered from Google Image search is 793 (Std. 99).

3.2 Features

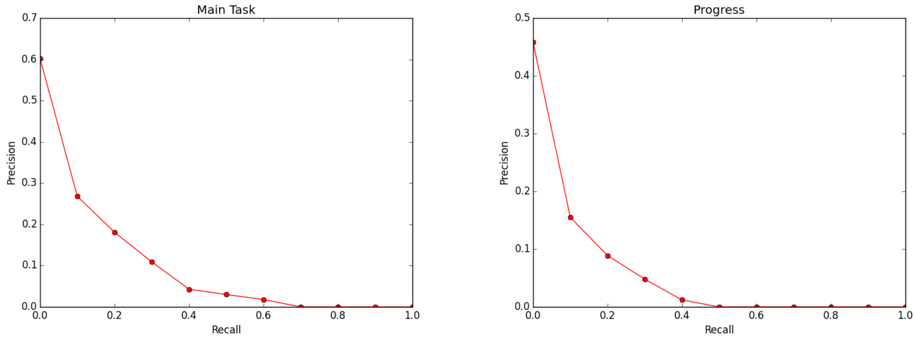
We used deep CNN global features for the SIN tasks. *Caffe* [7] is a popular implementation of deep neural network for computer vision. A pre-trained CNN network model from ImageNet [14] is used to calculate the forward propagation features. Not all activation values within the deep neural network are used, we choose 4096 dimension from the layer 7 from the CNN network to form our feature vectors. [10, 3] point out that the advantages of using features constructed from higher layer of Deep neural network in concept detection. Layer close to the input layer represent detail information such as edges, while layers close to output convey more semantic information.

3.3 Experiment and Results

We carried out experiments to train classifiers on TRECVid 2014 development dataset and classifiers using data gathered from external sources, namely Google Image Search and ImageNet. These external sources are search engines that retrieve images using textual queries. The classifier trained for main and progress task used data from the 2013x subset of the TRECVid 2014 development data and the classifier trained for no annotation task used external training data gathered as described in datasets. Performance of the classifiers were evaluated using inferred average precision (infAP). For no annotation task, classifiers for 34 of the 60 concepts were trained using data from ImageNet and the remaining using examples from Google Images. All classifiers trained using images from Google Images demonstrated poorer infAP than those trained on internal data. For no annotation task and main task the testing set is IACC.2.B, mean while progress task use IACC.2.C as testing set.

Figure 7 shows the mean inferred average precision (infAP) and recall curve of 30 test concepts. Based on the same linear SVM and extracted global CNN features, recall-precision curve of main task shows a superior performance than progress task.

When we compare the results of the main task (mAP = 0.086) and no annotation task (best mAP = 0.080), about 20 concepts has similar or better infAP score in no annotation task than the result in main



(a) main task

(b) progress task

Figure 7: Recall-Precision plot for Main and progress task

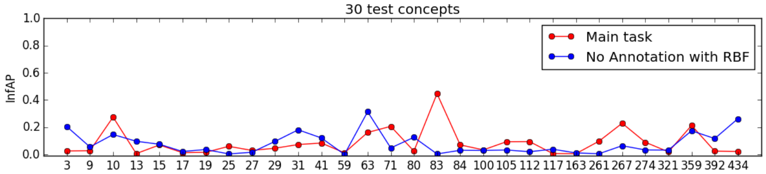


Figure 8: InfAP for 30 test concepts in no annotation task and main task

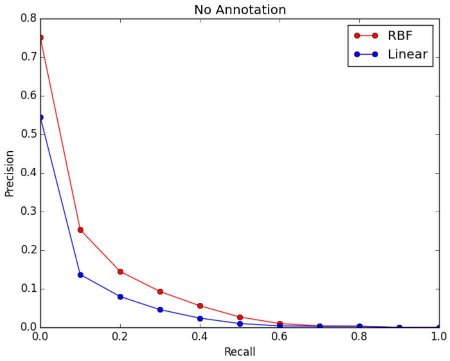


Figure 9: No annotation task with different Kernel in SVM

task. While the following concepts (Beach, Chair, Instrumental musician, News studio, Singing, stadium, flags, forest) have better score in main task.

The reason that the Main task out-performed no annotation task may be because there were fewer examples from ImageNet than in the TRECVid development set and in some cases the ImageNet examples were incorrect. For example, in the case of the concept "hand", several synsets matching the term consisted entirely of wristwatches. Finally, in other cases, the concept text (the query) was either ambiguous or insufficiently semantically rich, for example "greeting" (greeting cards were retrieved) and "government leader" (smaller subset of such leaders in internal training data).

4 Conclusions

In this paper we presented the participation of insight-dcu team in the INS and the SIN tasks. In the INS task, pre-trained deep convolutional neural networks (CNN) are used for feature extraction in three runs, where the matching is done by cosine similarity or by embedding the features in a Hamming space and use approximate nearest neighbor (ANN) indexes (best mAP = 0.13). These approaches shows the potential of using CNN and the need to improve the results by using user's annotations to train more sophisticated models to generate the scoring for the re-ranking. The forth INS run used a large visual vocabulary extracted from SIFT features, and spatial verification and query expansion to improve the matching performance (mAP of 0.12 is achieved), this approach motivate us to propose better Human detection mechanisms as well as considering noisy and cluttered background of some objects. For the SIN task, the potential of CNN is evaluated in the main task, the no annotation task and the progress task. The achieved results show close performance between the main and the no annotation task, and better querying mechanism to retrieve relevant training examples are necessary to enhance the results of the no annotation task. A possible solution is to use external lexical and semantic bases to build queries that reduce the false positive among the retrieved training examples.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under grant number SFI/12/RC/2289.
Programme material copyrighted by BBC.

References

[1] Rami Albatal, Kevin McGuinness, Feiyan Hu, and Alan F Smeaton. Formulating queries for collecting training examples in visual concept classification. In *Workshop On Vision And Language 2014*, 2014.

[2] R. Arandjelović and A. Zisserman. Multiple queries for large scale specific object retrieval. In *British Machine Vision Conference*, 2012.

[3] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. Neural codes for image retrieval. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 584–599, 2014.

[4] João Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV'12, pages 430–443, Berlin, Heidelberg, 2012. Springer-Verlag.

[5] Ken Chatfield and Andrew Zisserman. Visor: Towards on-the-fly large-scale object category retrieval. In *Computer Vision – ACCV 2012*, volume 7725 of *Lecture Notes in Computer Science*, pages 432–446. 2013.

[6] Lewis D Griffin. Optimality of the basic colour categories for classification. *Multimedia Tools and Applications*, 3.6:71–85, 2006.

[7] Yangqing J.

[8] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *Conference on Computer Vision & Pattern Recognition*, jun 2009.

[9] S. Kordumova, X. Li, and C. G. M. Snoek. Best practices for learning video concept detectors from social media examples. *Multimedia Tools and Applications*, pages 1–25, May 2014.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] Kevin McGuinness, Robin Aly, Ken Chatfield, Omkar Parkhi, Relja Arandjelovic, Matthijs Douze, Max Kemman, Martijn Kleppe, Peggy Van Der Kreeft, Kay Macquarrie, et al. The AXES research video search system. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014.

[12] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Quéenot. Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2014*. NIST, USA, 2014.

[13] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Berg A., and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014.

[15] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep fisher networks for large-scale image classification. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 163–171. Curran Associates, Inc., 2013.

[16] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[17] Alan F. Smeaton, Paul Over, and Wessel Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In Ajay Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*, pages 151–174. Springer Verlag, Berlin, 2009.

[18] Koen van de Sande, Theo Gevers, and Cees Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1582–1596, September 2010.

[19] Z. Zhang, R. Albatal, C. Gurrin, and A. Smeaton. Trecvid 2013 experiments at dublin city university. In *2013 TREC Video Retrieval Evaluation*, Gaithersburg, MD., 20-22 Nov 2013 2013.

[20] Liang Zheng, Shengjin Wang, Ziqiong Liu, and Qi Tian. Lp-norm idf for large scale image search. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1626–1633. IEEE, 2013.