

Automated Annotation of Landmark Images using Community Contributed Datasets and Web Resources

Gareth J. F. Jones¹, Daragh Byrne^{1,2}, Mark Hughes^{1,2}, Noel E. O'Connor²,
and Andrew Salway¹

¹ Centre for Digital Video Processing, School of Computing,
Dublin City University, Dublin 9, Ireland

² CLARITY: Centre for Sensor Web Technologies,
Dublin City University, Dublin 9, Ireland

{gjones,dbyrne,mhughes,asalway}@computing.dcu.ie, Noel.OConnor@dcu.ie

Abstract. A novel solution to the challenge of automatic image annotation is described. Given an image with GPS data of its location of capture, our system returns a semantically-rich annotation comprising tags which both identify the landmark in the image, and provide an interesting fact about it, e.g. “A view of the Eiffel Tower, which was built in 1889 for an international exhibition in Paris”. This exploits visual and textual web mining in combination with content-based image analysis and natural language processing. In the first stage, an input image is matched to a set of community contributed images (with keyword tags) on the basis of its GPS information and image classification techniques. The depicted landmark is inferred from the keyword tags for the matched set. The system then takes advantage of the information written about landmarks available on the web at large to extract a fact about the landmark in the image. We report component evaluation results from an implementation of our solution on a mobile device. Image localisation and matching offers 93.6% classification accuracy; the selection of appropriate tags for use in annotation performs well (F1M of 0.59), and it subsequently automatically identifies a correct toponym for use in captioning and fact extraction in 69.0% of the tested cases; finally the fact extraction returns an interesting caption in 78% of cases.

Keywords: web mining, geo-tagged images, landmark identification, automated image captioning

1 Introduction

Photo capture, storage and usage has undergone a revolution in recent years. Many people routinely take large numbers of images from their lives using dedicated digital cameras, and increasingly with those embedded in mobile devices such as smartphones. Many of the devices used for image capture now incorporate GPS sensors meaning that the location at which an image was captured is

easily available. These images are then very often shared with others using online photo archives such as Flickr or Facebook. Users uploading images are expected to provide captions for their images which must be entered manually. However, there are disadvantages with this manual annotation process. First, those taking pictures will often be traveling in places which they do not know well and so are not able to provide accurate and/or interesting labels. Thus images taken while on a visit to London may simply be labeled “London”. The volume of images taken on such a trip means that even if they are knowledgeable about the place being visited, users will often not take the time to provide detailed captions, and even if they do this, labels will be inconsistent between different users uploading to social repositories, reducing the effectiveness of subsequent image search. In this paper we describe a mobile application running on an iPhone in conjunction with a web service which automates this captioning process for landmark images. Once captioned images can be uploaded to an online social media application. We also believe our work is relevant in the context of the burgeoning interest in augmented reality whereby a camera screen on a mobile device is automatically supplemented with caption-like details about the target image.

Our approach exploits GPS information accompanying an image, geographic resources that provide reverse lookup, e.g. GeoNames [1], the existing keyword tags associated with images in community contributed datasets such as Flickr, and the information about a great many places available taken from the World Wide Web. Given a GPS-tagged image of a landmark our system can generate a caption comprising keyword tags that describe what landmark is in the image and give an interesting fact about it. Classification is achieved through the integration of image classification based on computational classification methods (using local image features [4] and Support Vector Machines [6]) and a technique for text information extraction from the web.

This paper is organised as follows: Section 2 reviews background to our work, Section 3 describes the component stages of our system and how they are integrated to generate image captions, Section 4 describes the application and evaluation of the components, and finally Section 5 concludes our work to date.

2 Background

The ‘semantic gap’ between the low-level features used in many content-based image analysis technologies and the human interpretation of images means that most large-scale image retrieval systems in use today are based on textual metadata in the form of tags or captions which are usually created manually by humans. Thus the potential for retrieved images to satisfy the user relies heavily on the accuracy and quality of these human-defined tags. The main disadvantage of this approach is that these manual tags will be inconsistent and often poor due to lack of knowledge or time on the part of the annotator.

An alternative and more appealing image search scenario in many situations is to use a query image which should be compared to existing known images in order to identify what is depicted in the new image. High-level semantic

classification can be used to identify complex images such as the image being a view of the Trevi Fountain in Rome. A popular approach to automatic high-level semantic classification is to use local image features as opposed to low-level image features. Local image features are based around interest points (salient, non uniform regions) in a photo. These have been successfully used in the past for object matching, tracking and recognition along with other niche tasks such as image mosaicing [9]. Several research groups have used these local image features for object recognition tasks on mobile devices. For example Chevallet et al. [5] developed an application called Snap2Tell designed to run on GPS enabled mobile phone devices. In their system a user can take a picture of one of 120 landmarks located around Singapore (STOIC dataset), which is then identified on a remote server. Fritz et al. [7] use SIFT features for descriptor matching in a mobile landmark recognition system. They use a relatively small dataset of 1005 landmark images (ZUBUD dataset) based around Zurich as their training collection. When a user takes a picture of a landmark within the Zurich region, the system aims to classify the image against the dataset using only the SIFT features that it deems to be ‘informative’, disregarding all other features to speed up matching time. Yeh et al. [19] developed a system that recognises an image of a landmark taken on a mobile device and retrieves information from an online search describing the landmark within an image. This system compares an image of a landmark taken on a mobile device against a collection of images that are contained on webpages. If a match is found, the system extracts information from the text in the corresponding webpage and uses it to extract information from the wider web to describe the landmark depicted within the image.

Our work improves on this previous work by implementing a more robust recognition system that is able to classify landmarks more accurately using very large datasets of community contributed images. Our techniques allow for the creation of reliable captions and tags from large amounts of noisy data. The integrated framework then reliably augments these captions and tags with facts about landmarks contained within an image taken from the whole Web.

3 Landmark Identification and Caption Generation

Our landmark identification and captioning application exploits the very large number of captioned geo-tagged images now available in community contributed image collections from which tags are selected. In this integrated process when a new geo-tagged image is introduced into the system, a cluster of similar images is first identified. Representative sets of tags for this landmark based on a set of matched images are then identified and used to identify the primary toponym in the image. Finally keywords selected from images related to the one being captioned are used to extract a fact about the landmark from the web and integrate it into the image’s caption.

3.1 Landmark Classification

An effective approach to classification of a landmark image is to harvest a large number of similarly annotated landmark images, and then to match it based on context and content features of these images [14] [15]. In this process image and object matching using interest point features has been shown to work well even in large-scale image databases containing thousands of different images [11]. However the actual matching between keypoints can be very computationally expensive, and for large-scale image databases containing millions of images computationally infeasible. In order to use these techniques in a practical system, methods are required to reduce the number of keypoints which need to be compared or else that do not match keypoint to keypoint. In our work this is achieved by combining computer vision techniques with different forms of semantic context data to organise and classify landmarks within images.

A new framework is implemented based on *single viewpoint clustering* [10] which enables the efficient and accurate classification of landmarks using a large scale training database. Single viewpoint clustering involves collecting a number of images of the same landmark taken from a relatively similar viewpoint and clustering them to create clusters of visually similar images. Each cluster can then be assigned spatial location data. They can be used for efficient classification of new input images using Support Vector Machines (SVMs), where each SVM model represents a single landmark from a certain viewpoint. One drawback of this approach is that a large number of positive examples are needed to train an accurate SVM classifier. Within real-world collections of landmark images, there may not be a sufficient number of images to build a reliable SVM model. To deal with this situation, we use a method that addresses this problem by combining SVMs with an hierarchical classification approach.

In this paper we apply this technique to a collection of images harvested from Flickr which contains many examples of images for significant landmarks. For example a search on Flickr for “eiffel tower” currently returns over 370,000 images and for “notre dame paris” over 245,000 images (May 2010). Landmarks tend to have a unique visual appearance that leads to high discrimination values between different landmarks. The automated classification approach applied here works well due to the observed capture behaviour of users on large scale photo-sharing websites. Photographers tend to visit similar destinations and landmarks, and to take images of these landmarks from a small number of locations due to geographical constraints and their photogenicity from certain viewpoints. This leads to a large overlap of visually similar images of popular landmarks. Based on this observation, our system takes advantage of this overlap by reducing the search space in a large scale dataset by clustering similar images, thus creating a robust means of classifying an image using SVMs.

Dataset For this work a training collection of images was harvested using the Flickr API from the metropolitan area of Paris. In order to reliably cluster similar images for SVM training, we downloaded only geo-tagged images. To ensure that the vast majority of these images contained large landmarks as their main

subject, the image text tags assigned to describe it when it was uploaded to Flickr were analysed. A long list of stopwords was created to filter out unwanted images (eg. concert, match, march). This filtering process produced a training set of 76,749 geo-tagged images. However, since geo-tags and text tags are assigned by those contributing the images to Flickr there is no way to ensure their accuracy. It is however expected that correct tags will dominate the dataset.

SVM Classification Single viewpoint clustering involves taking a number of images of the same landmark taken from a relatively similar viewpoint, and clustering them into visually similar clusters. Due to the large size of the dataset, traditional clustering techniques such as K-means would be infeasible. We explored many different techniques and combinations of image features (content and context) to determine an efficient and accurate method to cluster large numbers of images. We selected a combination of spatial data, low-level image features and SURF local image features [4].

All training images were first clustered based on geographical locations. Cluster centres were chosen randomly in the dataset and all images located within a spatial radius of 500 metres of each centre were clustered. This process was repeated until all images in the dataset were assigned to a cluster. Within each of these clusters, images were then subclustered based on two MPEG7 low-level features: edge histogram (which provides weak spatial verification) and scalable colour using an hierarchical clustering method.

Each of these clusters were then subclustered again based on local image feature matching. A graph was created using images within a cluster as the nodes and local feature matched as the edges. The most connected image was chosen as the cluster centre. All images were then compared against the cluster centres to subcluster these images into visually similar clusters. Thus, each cluster should represent a landmark from a similar viewpoint. Clusters were then assigned a spatial location based on the average position of each image within the cluster.

K-means clustering was then carried out on these clusters ($K = 100$) using their geographical location as the comparison values. The metropolitan area of Paris was split into 100 geographical regions and a multi-class SVM model trained to represent all classifiable landmarks within each of these regions. Thus 100 multi-class classification models were trained in total, each one representing all viewpoints of landmarks within the geographical bounding box as determined by the K-means clustering procedure. These models were trained using Visual Bag of Words (BOW) features with a vocabulary size of 4096. We also trained versions of these SVM models using the MPEG7 edge histogram descriptor.

To classify an image, firstly its closest multi-class SVM is retrieved based on geographical distance. A visual BOW is created for the test image based on the vocabulary used for the training images. The SVM then classifies the test input vector into one of the classes used to create the model. At this point an input image is only classified to the nearest class so a more definite verification is required to guarantee an accurate match. The input image is then compared against all images within this class using point to point matching with SURF

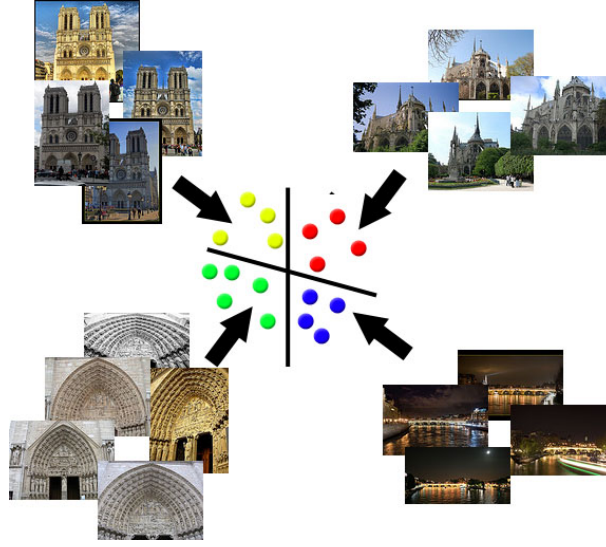


Fig. 1. Example of the SVM training process within a small spatial area in Paris. All clusters within this geographical area are used as inputs into a multi-class SVM model.

image features. If the number of matches is above a threshold, this should confirm that the input image is indeed a match.

A significant drawback of this approach is that a large number of training examples are needed to build each accurate SVM model. Our hybrid approach addresses this problem as follows, in cases where no match is found for an input image using the SVM approach, a slower hierarchical pipeline classification method is used.

Hierarchical Classification To classify an input image using the hierarchical approach all images within a spatial radius are first retrieved. Many different spatial radii were investigated. It was found that for our urban Paris dataset that a radius of 500 meters provided the best trade-off between accuracy and speed. Although high level semantics based on image features are very difficult to implement successfully, several low-level semantic classifiers can work quite well. In our work two low-level semantic classifiers that have been shown to work well in the past are indoor/outdoor and building/non-building [17][12], were used in the early phase of our hierarchical pipeline. Classifiers were trained based on MPEG7 features using SVMs to classify whether an image was taken indoors or outdoors and whether an image contains a large building. The number of retrieved images is then pruned based on the results of these semantic classifiers. Gabor texture features are then extracted and compared against the remainder of the retrieved images. All images which have a Euclidean distance above a threshold (threshold = 20) are then pruned from the search space. Point to point matching is then carried out using SURF image features and the distance

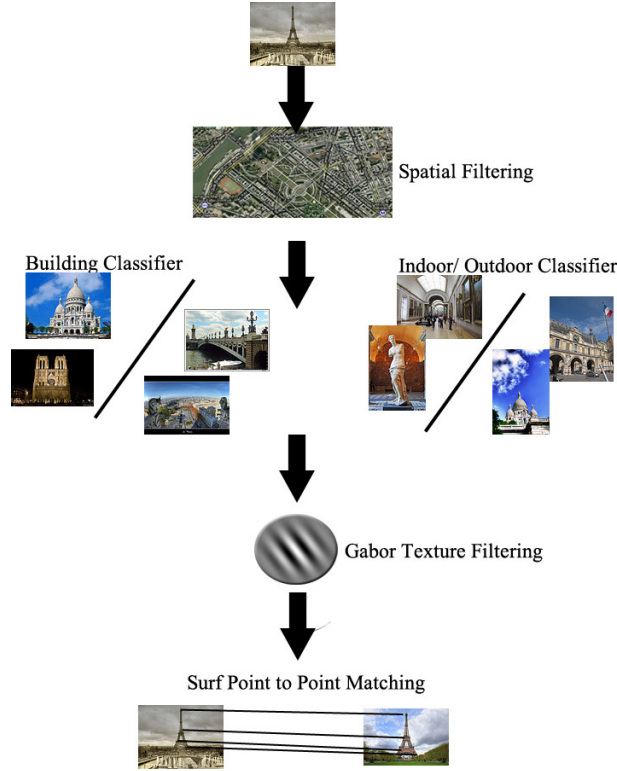


Fig. 2. The hierarchical classification pipeline. Images that cannot be successfully classified using the spatially organised SVM models are classified using this approach.

ratio test. The image with the highest number of positive matches (if above a threshold of 4) is then matched with the input image

3.2 Landmark Identification and Selection of Keywords, Tags and Image Title

The landmark image localization and matching method described in the previous section was implemented into a matching engine and service. The engine localizes and matches the input image, looking up potentially relevant images within the training dataset of Parisian landmark images. Once potential matches have been located, the engine returns a set of images from the training set ordered from best match to worst in XML format. For each image returned, associated metadata is also provided, including the community-contributed tags, the title given to the image by the uploader and a unique identifier for the matched image.

Tag Filtering And Selection Each matched image returned by the localization engine has associated information which includes a set of tags. We wish

to assign the most appropriate tags to the input image from among those contributed by the community. Thus our system attempts to identify a representative set of tags which can be used to annotate the target image. A hashmap of tags is created from the available tag set, by iterating through the matched images and the tags they contain and adding them to the set. A tag within the set is given an importance measure or weighting which is incremented with each encounter. The amount by which it is incremented corresponds to the rank of the matched image within the results list, i.e. tags encountered in the highest scoring match are given greater weight than those from images further down the list. Finally, using the weighted score of each tag, the set is thresholded to yield the set of tags likely to be the most representative of the target image.

Toponym Identification Using the provided location information, our application middleware communicates with the GeoNames API and retrieves a list of toponyms within a 1.5 kilometre radius of these coordinates. A wide radius is employed to allow for positioning errors or varying ranges of accuracy in the provided GPS position. The toponym list is filtered to remove irrelevant, spurious or noisy toponym types, e.g. names of hotels. The list of potential toponyms is then matched as outlined above to the selected tags to identify the most likely candidate. This is then applied as the image title. With a toponym identified, the fact extraction and caption augmentation service, as described below, is called and the returned facts are then used to further annotate the media being processed. We employ a reasonably straightforward approach: the available toponyms are compared to the thresholded tag set and the best matching item chosen. Stop-words are removed from the titles, which are divided into tokens and stemmed using the Porter Stemming algorithm [13]. Using the Jaccard Coefficient [18], each toponym is then compared to the selected tags, and scored. The best match is then returned, or if no match is found, the closest toponym is returned.

3.3 Fact Extraction and Title Augmentation

In the next stage of processing we use the output of the image classification and toponym identification as input to a highly portable mechanism for the extraction of partially structured facts from information on toponyms available on the World Wide Web. A particular feature of this is that it exploits information redundancy on the web, i.e. the fact that the same information about a landmark is available in many forms on the web. This method is described in detail in [16]. For a given landmark, we return a list of facts in the form (Landmark, Cue, Text-Fragment), ranked according to a score which is intended to promote interesting and true facts. This fact structure makes it straightforward to combine it with an existing image title. Crucially, for this information extraction process, we assume that at least one key fact about a landmark will be expressed somewhere on the web in a simple form, so that we only need to work with a few simple linguistic structures and shallow language processing. The following sub-sections describe the fact extraction process.

Get Snippets from Search Engine: A series of queries is made to a web search engine (we use Yahoo’s BOSS API [3]). Each query takes the form <“Landmark Cue”>; where the use of double quotes indicates that only exact matches are wanted, i.e. text in which the given landmark and cue are adjacent. A set of cues is manually specified to capture some common and simple ways in which information about landmarks is expressed, e.g. ‘is a’, ‘is famous for’, ‘is popular with’, ‘was built’.

Although we worked with around 40 cues (including single / plural and present / past forms), a much smaller number are responsible for returning the majority of high ranking facts; in particular (and perhaps unsurprisingly) the generic “is” seems most productive. The query may also include a disambiguating term. For example, streets and buildings with the same name may occur in different towns, so we can include a town name in the query outside the double quotes, e.g. <“West Street is popular with” Bridport>. For each query, all the unique snippets returned up to a preconfigured maximum number are processed in the next step. Typically a snippet is a few lines of text from a webpage around the words that match the query, often broken in mid-sentence.

Shallow Chunk Snippets to Make Candidate Facts: Because we are only retrieving information about a given landmark that is expressed as “Landmark Cue ...”, we can use a simple extraction pattern to obtain candidate facts from the retrieved snippets. The gist of the pattern is ‘BOUNDARY LANDMARK CUE TEXT-FRAGMENT BOUNDARY’, such that ‘TEXT-FRAGMENT’ captures the ‘Text-Fragment’ part of a fact. The details of the pattern are captured in a regular expression on a language-specific basis, e.g. to specify boundary words and punctuation, to allow optional words to appear inbetween LANDMARK and CUE, and to reorder the elements for non-SVO languages. A successful match of the pattern on a snippet leads to the generation of a candidate fact. For example, using extraction patterns the snippet text ‘...in London. Big Ben was named after Sir Benjamin Hall. ...’ matches, giving the candidate fact (Big Ben, was named, after Sir Benjamin Hall) but ‘The square next to Big Ben was named in 1848...’ does not match.

Filter Candidate Facts: Four filters are used as a quality control to remove candidate facts that: contain potentially subjective words; end in words that would be ungrammatical; are under a length threshold; and that contain words that are all in capitals. Finally, facts are ranked so that we are more likely to get correct and interesting facts at the top. We exploit the overlap between candidate facts for the same Landmark-Cue pair to capture these notions to some extent. For each Landmark-Cue pair a keyword frequency list is generated by counting the occurrence of all words in the Text-Fragments for that pair, words in a stopword list are ignored. The score for each fact is then calculated by summing the Landmark-Cue frequencies of each word in the Text-Fragment, so that facts containing words that were common in other facts with the same Landmark-Cue will score highly. If shorter facts are wanted then the sum is divided by the word length of the Text-Fragment.

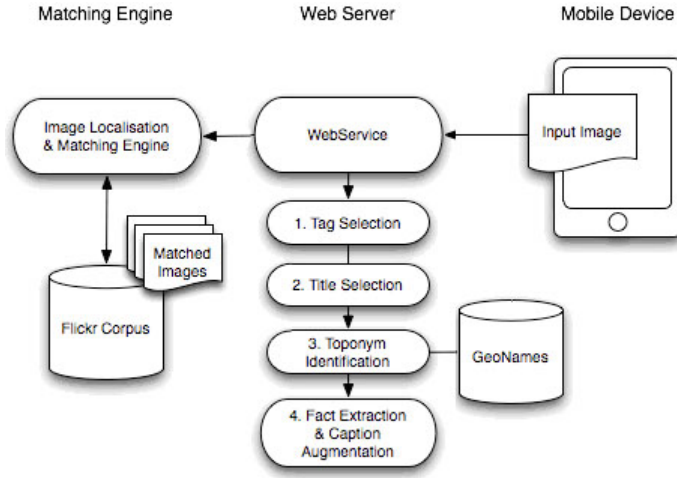


Fig. 3. An illustration demonstrating how the different components of the framework integrate with one another.

The sum score for a fact can become high in two ways: (i) there are many overlapping Text-Fragments for an Landmark-Cue pair, so there are some high word frequencies; and (ii) a fact contains more of these high frequency words than other facts. Thus, the method is designed to highly rank facts with the most appropriate Cue for the Landmark, and the best Text-Fragment for the Landmark-Cue pair. For an existing image title, e.g. “A view of the Eiffel Tower”, then the top-ranked fact, e.g. ‘Eiffel Tower, was built, in 1889 for an international exhibition in Paris’, can be inserted in one of two ways: (i) as a new sentence - “A view of the Eiffel Tower. The Eiffel Tower was built in 1889...”; or (ii) as a subclause - “A view of the Eiffel Tower, which was built in 1889...”.

4 Application and Evaluation

4.1 Application Workflow

The three components described in the previous sections are integrated into a combined service architecture shown in Figure 3. The landmark image recognition and classification engine resides on the server along with a series of web services designed to expose their functionality to a mobile application running on a compatible mobile device (in this case an iPhone.) The integrated service allows an input landmark image to be recognized, localized and matched with other images in the repository, in this case the Flickr corpus of Parisian landmarks described previously. In order to caption and tag an input image, matches for the provided image are looked up, and through the middleware layer used to determine appropriate annotations to be applied to the target image. The steps are as follows: first the image matching is performed after which, and using



Input Image 	
1. Image Matching Results	
2. Selected Tags	france, louvre, paris,
3. Identified Toponym	Louvre (Distance = 0.2721)
4. Returned Facts	<p>Louvre. Louvre was built on the site of a medieval fortress on the banks of the Seine river.</p> <p>***</p> <p>Louvre. Louvre is famous for the Mona Lisa painting.</p> <p>Louvre. Louvre was built as a residence for the kings of France.</p>

Fig. 4. The workflow and outputs of the chained components illustrated with a worked example, in this case of the Louvre in Paris.

the returned results along with their associated metadata, a set of representative tags for the image being processed is identified. Using these tags, the best matching toponym nearby the provided coordinates is determined, this is then used to seed the fact extraction and caption augmentation step. This workflow is illustrated using a worked example in Figure 4.

4.2 Mobile Application

The mobile application is designed to operate in-situ with a tourist style scenario in mind. Examplescreenshots of the working iPhone application are shown in Figure 5. The application operates as follows: first the user selects a photo they want to process, either by taking a new image with the device's in-built camera or by selecting an existing image from the photo library. They are then asked to confirm that the location for the image is correct, after which the image and location data is passed to the middleware layer through a REST-based API. After the service completes the matching and annotation of the image, a response is returned to the device. The annotated image is then saved to a local data store and the application presents the results on-screen. The image, along with the automatically generated captions and tags, can then be uploaded to a number of social media sites including Flickr and Twitter through the results screen.

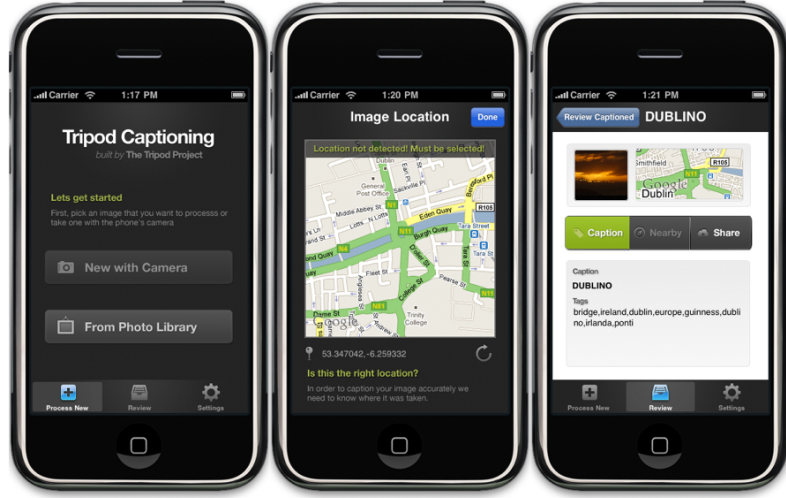


Fig. 5. An example of the application developed based on this framework running on an Apple iPhoneTM mobile device.

Table 1. Landmark Classification Accuracy(270 test images)

Approach	No. of images	Classified correctly
Hierarchical only	270	91.0%
SVM (BOW) only	156	92.9%
SVM (Edge) only	214	93.4%
Hybrid (BOW)	270	93.3%
Hybrid (Edge)	270	93.6%

4.3 Evaluation

This section describes laboratory evaluation of the accuracy of the components of our landmark image classification and annotation system.

Landmark Classification In order to evaluate the classification system, a test collection of 270 images was gathered from the Panoramio online collection using Panoramio’s REST API [2]. All of these test images have a large landmark as their main subject and include geo-tags. It should be noted that to make the test collection realistically challenging, the landmarks shown in many of the images are partially occluded and taken under a wide variety of lighting conditions (day, night, flash, etc.). The test collection was first classified solely using the hierarchical classification approach described in section 3.1, with the aim of ascertaining the classification accuracy of this technique. Experimental results are shown in Table 1.

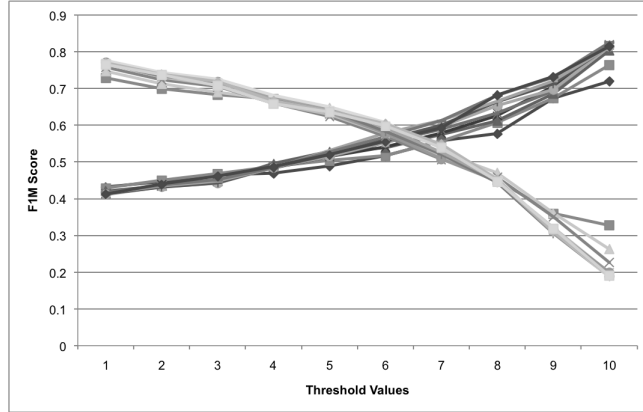


Fig. 6. The precision can be seen to increase for each of the weighting approaches as the thresholds increase, while the recall increases.

We then processed our training set of images into spatially organised multi-class SVM models (using BOW and Edge features) and re-classified the test collection using the hybrid approach combining SVMs and the hierarchical approach. Results of this investigation are again shown in Table 1. Of the 270 images within the test collection, the BOW SVMs recognised 156 of them while the Edge based SVMs recognised 214 of the test images. The Edge based hybrid approach slightly outperforms the BOW using this dataset.

Tagging The selection of appropriate tags for the target image is extremely important within the workflow since tags are used for image annotation and as the input to the web-based augmentation stage. In order to evaluate the accuracy of the tagging phase, we groundtruthed 85 test images. The tags for the images returned for each image by the image matching service were formed into a union pool set. An annotator manually judged each pooled tag and made a binary classification of its relevance (ie. relevant or not). A tag was determined to be relevant if it described the landmark featured in the image. As such associated tags were not deemed relevant. Thus details such as descriptions of camera types, related tags such as weather or lighting, or information on the year or events and activities were deemed non-relevant, since the emphasis in the groundtruthing was placed on tags which identify the landmark in the image. On average 9.85 tags were deemed relevant per image, while each image had an average of 69.09 tags taken from 12.85 images matches from the image collection.

With a groundtruth established, the tag weighting and thresholding approach as previously described was applied to the image matching results for each test image. The input parameters were varied from 0.5-0.95 for both weighting and threshold and all of the combinations iterated. The set of tags returned from each variation was compared against the groundtruth for that image. Precision and recall measures were calculated as outlined in [8]. These were then averaged across all of the test images and the F1M measure calculated.

Within the selection of tags there was a need to balance precision and the recall so that ‘noisy’ or superfluous tags are kept to a minimum while a maximum of the desired tags are contained within the selected set. Applying a low threshold results in an unconstrained and highly noisy set displaying high recall but extremely low precision. This is illustrated in Fig 6. Conversely, a higher threshold and weight results in an overly constrained set, which while displaying high precision, has very low overall recall. By exploring the various variations for the highest F1M, we identified a threshold value of 0.75 with an iteratively decreasing tag weighting of 0.85 to be optimal for tag selection. This resulted in on average 13.9 tags being selected. While some of the selected tags are not directly relevant, this may be acceptable to users since many of these additional tags are noted to be ancillary or related descriptors rather than genuine noise.

Toponym identification A toponym is used to initiate the fact extraction and title augmentation step, and its accuracy is thus important to the effectiveness of the fact extraction stage. The identification of the appropriate toponym label for each image is reliant upon the outputs of the tag filtering and selection process as outlined previously. To investigate this, 87 test images were selected, image matching was performed and a set of tags filtered from the results selected. Nearby toponyms were looked up using GeoNames and a candidate selected based on the tag set in a manner as outlined in Section 3.2. Each of the returned toponyms was then annotated into one of the following categories: Incorrect toponym identified; Vague or unspecific toponym identified, e.g. Paris, France; Toponym is related to the target but is incorrect, this included a landmark nearby or within the image but which was not the primary focus or featured landmark, e.g. the Champ de Mars returned in place of the Eiffel Tower; and finally a correctly identified toponym. In total 13 of the toponyms were incorrect, 4 were vague, 10 were incorrect but related and 60 were correct.

While 68.97% of the tested images returned a correctly identified toponym, a further 16% (vague and related categories) may be considered acceptable (totalling 88.5%). All of the vague cases were composed of a generic toponym of ‘Paris’, which returned facts such as ‘Paris is named after a Celtic tribe called the Parisii who lived on the island in the river’, ‘Paris is famous for its huge number of cafes and brasseries’ and ‘Paris was made for lovers and lovers of life’. While these facts are not ideal, they are generic enough to be reasonably acceptable. Additionally those which are related often contained reference to the target landmark. For example, in the case where the Champ de Mars was identified in place of the Eiffel tower, the first returned fact is the following: ‘Champ de Mars is a green area located in the middle of the Eiffel Tower and the Ecole Militaire building’. In another case, where the Champs Elysees was returned instead of the Arc de Triomphe, the first fact returned again referenced the desired landmark: ‘Champs-elysees is a seventeenth century garden-promenade turned avenue connecting the Concorde and Arc de Triomphe’.

Fact Extraction and Title Augmentation For this evaluation 68 place names from around Europe were selected. We chose an even mixture of urban

/ rural and famous / not famous places from European cities (London, Riga, Zurich and Dublin) and countryside (UK, Latvia, Switzerland and Ireland), and various types of place - churches, statues, mountains, rivers, etc. For each place the top ranked fact was evaluated in terms of its correctness (according to one investigator consulting relevant websites) and whether or not it was deemed interesting (according to the judgments of five subjects). Our evaluation criteria were actually rather strict, since it was found that a majority of subjects rated more facts as ‘interesting’ (78%) than we ourselves rated as correct (50%).

Overall we are encouraged by the performance of each of the component technologies used in our application. While all of them use some degree of empirical design and parameter selection, none of these specifically focus on the dataset used here and they can easily be adjusted for other similar environments. We anticipate that in an operational application empirical parameters could be adjusted automatically based on user feedback.

5 Conclusions and Further Work

This paper has described our novel integrated system for automatically captioning landmark images captured on a GPS enabled mobile device. Evaluation of the three principle components of the systems shows that they each have a high degree of effectiveness. They do however make some mistakes. However, informal testing of the combined system shows that the application provides good tags for popular landmarks for which there are many existing labeled images contained in online social collections with a correspondingly large number of facts available online. Tagging is less effective for less frequented landmarks where there are less online images and less web content available. For this latter case we plan to explore more sophisticated techniques to improve the quality of tagging, however it is likely that for many less popular landmarks this problem will address itself over time as more online content appears naturally. We also plan to carry out an end-to-end evaluation of the captioning system. This will evaluate the accuracy of landmark identification, factual augmentation, and the acceptability and value of the captions to users.

In the longer term, users will be able to upload captioned images to social image collections. Over time this will expand the number and detail of annotated images available. This itself will provide more effective sources of information for landmark identification and for the selection of accurate and interesting tags. While the system described here is only implemented for English, the methods used are all language independent and it could easily be ported to other languages by means of new stopword lists, localised toponyms lists and collections of word patterns in the fact extraction stage [16].

Acknowledgement

The research reported in this paper is part of the project TRIPOD supported by the European commission under contract No. 045335.

References

1. Geonames. <http://www.geonames.org>.
2. Panoramio. <http://www.panoramio.com>.
3. Yahoo! search boss. <http://developer.yahoo.com/search/boss/>.
4. H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision*, pages 404–417, Graz, Austria, 2006.
5. J.-P. Chevallet, J.-H. Lim, and M.-K. Leong. Object identification and retrieval from efficient image matching. Snap2Tell with the STOIC dataset. *Information Processing and Management*, 43(2):515–530, 2007.
6. C. Cortes and V. Vapnik. Support-vector networks. (3):273–297, 1995.
7. G. Fritz, C. Seifert, and L. Paletta. A mobile vision system for urban detection with informative local descriptors. In *Proceedings of the IEEE International Conference on Computer Vision Systems (ICVS 2006)*, page 30, 2006.
8. R. Jäschke, F. Eisterlehner, A. Hotho, and G. Stumme. Testing and evaluating tag recommenders in a live system. In *Workshop on Knowledge Discovery, Data Mining, and Machine Learning*, pages 44–51, 2009.
9. F. Lorenz Wendt, S. Bres, B. Tellez, and R. Laurini. Markerless outdoor localisation based on sift descriptors for mobile applications. In *Proceedings of the 3rd international conference on Image and Signal Processing (ICISP 2008)*, pages 439–446, 2008.
10. D. G. Lowe. Local feature view clustering for 3D object recognition. volume 1, page 682, Los Alamitos, CA, USA, 2001. IEEE Computer Society.
11. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
12. J. Malobabic, H. le Borgne, N. Murphy, and N. O'Connor. Detecting the presence of large buildings in natural images. In *Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing (CBMI 2005)*, pages 529–532, 2005.
13. M. F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.
14. G. Qingji, L. Juan, and Y. Guoqing. Vision based road crossing scene recognition for robot localization. In *Proceedings of the International conference on Computer Science and Software Engineering*, volume 6, pages 62–66, 2008.
15. R. Rahmani, S. A. Goldman, H. Zhang, S. R. Cholleti, and J. E. Fritts. Localized content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1902–1912, 2008.
16. A. Salway, L. Kelly, I. Skadina, and G. J. F. Jones. Portable extraction of partially structured facts from the web. In *Proceedings of the 7th International Conference on Natural Language Processing (IceTAL 2010)*, pages 345–356. Springer, 2010.
17. M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *Proceedings of the IEEE International Workshop on Content-Based Access of Image and Video Database*, pages 42–51, 1998.
18. C. van Rijsbergen. *Information Retrieval (2nd edition)*. Butterworths, 1979.
19. T. Yeh, K. Tollmar, and T. Darrell. Searching the web with mobile images for location recognition. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 2, pages 76–81, 2004.