2023

Text-Based Guidance for Improved Image Retrievalon Archival Image Dataset

Ian Comor
University of Wollongong

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

TEXT-BASED GUIDANCE FOR IMPROVED IMAGE RETRIEVAL ON ARCHIVAL IMAGE DATASET

A Thesis Submitted in Partial Fulfilment of
the Requirements for the Award of the Degree of

Master of Philosophy (Computer Science)

from

UNIVERSITY OF WOLLONGONG

by

Ian Comor

School of Computing and Information Technology
Faculty of Engineering and Information Sciences

2023

Declaration

I, Ian Comor, declare that this thesis is submitted in partial fulfilment of the requirements for the conferral of the degree Master of Philosophy (Computer Science), from the University of Wollongong, and the work is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Abstract

Digitised archival photo collections allow members of the public to view images relating to history and democracy. Recent advancements in visual tasks such as Content Based Image Retrieval and the development of deep neural networks have provided modern methods to analyse digitised images and perform image queries for retrieval. We explore the image retrieval task using several publicly available datasets, and a set of archival images from the National Archives of Australia, and propose a simple change to existing pooling method to improve retrieval performance in the archival set.

Another visual task of object localisation considers the ability of a model to be trained to adequately locate in an image the positions of objects, given English text phrases. With other recent advances in large-scale text embedding models, pre-trained text models retain rich semantic structure within them. While other methods of object localisation involve the training of text pathways in their deep neural model, we explore direct use of a large-scale text embedding for this task, and demonstrate its ability to localise objects, and even on unseen words.

With our aim to continue the enhancement of retrieval performance of difficult archival datasets, we question if such a pre-trained localisation model can improve archival retrieval performance, due the nature that the archival set contains text information that can be harnessed.

We find modest improvement on the retrieval task using a trained localisation model that exploits the rich semantic structure of an off-the-shelf pre-trained word embedding model. This is promising in that the use of text-guided localisation can be an integral part of future archival image dataset retrieval.

KEYWORDS: Image Retrieval; Content Based Image Retrieval; Text Guidance; Visual Grounding; Object Localisation; Word Embeddings; Convolutional Neural Networks

Contents

List of Figures

List of Tables

Chapter 1

Acknowledgements

I would like to thank my primary supervisor Prof. Lei Wang for his years of unending patience during the time of my study. Without his guidance, advice, and tolerance this would not be at all possible.

I would also like to thank Dr. Peng Wang for much needed technical assistance.

During this degree I had the pleasure of meeting many brilliant students, including Zhongyan Zhang, Saimunur Rahman, Biting Yu, Din Sangrasi, Yu Ding, and Zhexuan Zhou.

However, I had the enduring pleasure of countless hours of study and synergy with fellow student Bela Chakraborty, where many late nights were spent learning together, bouncing ideas off one-another, or tackling inevitable coding catastrophes as they arose. Her energy and drive are unmatched.

Chapter 2

Introduction

Background

Image Retrieval

The proliferation of smartphone cameras, and the advent of low-cost remote sensing and high-speed Internet means large-scale image galleries can be produced inexpensively. Smartphone galleries can hold many thousands of photos, while photos can be backed up on personal computer systems.

Large-scale galleries can include Google Streetmaps, digital photo albums, social media photo collections, satellite images and remote sensing [55], academic datasets [24], and digitised historical images. The difficulty in a user retrieving a specific image or images from datasets containing thousands, or even millions, of images becomes apparent.

Image Retrieval (IR) is the automated process of retrieving images from an image database (a gallery) by descending order of their similarity to a user's query without the need for exhaustive manual searching. Retrieval should be performed using a human-friendly query, whether it be via keyword text input, an example image, or some other novel means. This should be possible without expert knowledge or onerous search fields, or the repeated finetuning of past queries.

Finding a specific photo requires onerous user browsing via exhaustive scrolling, which is not user-friendly. As a user's digital photo collection increases in size, it becomes difficult to navigate and manage. Large image collections may even compel the user to manually split the gallery by event or location by producing file subdirectories, which takes time and mental effort. A more human-friendly system would react to human queries, such as "Show me pictures of my holiday to Europe", "Show me all my photos with my dog", or "Show me all the pictures I took of the Eiffel Tower". Such a system could also sort the collection by similarity to one particular photo the user is interested in.

Advanced IR systems even have potential in commercial applications. With a sophisticated IR system journalists and historians could easily find images and footage for use in reporting [30, 105], art curators could easily search and explore artistic works, and environmental scientists can classify remote sensing images according to the presence of specific features. The potential applications of IR span into the fields of criminal investigation, medical imaging [81], remote sensing [106], astronomy [21], history, and heritage.

The intention of IR is to retrieve images by such means: using text and visual data. Broadly, IR can be divided into Text-Based Image Retrieval (TBIR) and Content-Based Image Retrieval (CBIR).

Text-Based Image Retrieval

Text-Based Image Retrieval (TBIR) [63] compares the user's text query with the text associated with each database image. It relies on three important factors: (1) the user to input a useful query representing their intention, (2) images to be sufficiently annotated with text that users will search for, and (3) a system (algorithm) that parses the texts so that relevant images are retrieved. Text annotations and descriptions alongside images allow for text-based retrieval. It is historically more common than CBIR because it is a facsimile of text document retrieval. Document retrieval is both easy to implement using already-existing algorithms, and is computationally lightweight. However, the implementation of TBIR introduces a broad

range of problems in the annotation process: the production of comprehensive and useful metadata can come at considerable expense, may require the knowledge and experience of experts in the relevant field, and background knowledge of the archival material.

Challenges in Text-Based Image Retrieval

The use of a text-based retrieval system with images requires that all images be accurately described with text descriptions. This can be a prohibitively laborious, and therefore expensive [91], requiring workers to research and cross-check before committing their annotations via data-entry. Furthermore, information about the image may be unavailable [109], unsuitable, or even incorrect. Image collections on a niche subject would require annotation by a related expert with background knowledge [14, 40, 91] or a relevant professional [125] to articulate the details, and this may not be possible [30, 88, 111]. Organisations on strict budgets may 'cut corners' and apply bulk descriptions to multiple images according to location, date, or photographer. The annotations may also be encumbered with subjective bias of the annotator or local colloquialisms [88], or may not match common user queries. Furthermore, relevant experts, even if available, may annotate with jargon terms that laypersons may not know.

Text details can be on a higher semantic level to the actual visual content [91], meaning historical images may be described in a different manner to other types of images [30]. For example, annotations may describe the situation behind an event ("city mayor marks the opening of a new bridge") rather than the actual visual content ("man cutting ribbon with large scissors").

For the user's part, they may not know what they are looking for, or lack the vocabulary to adequately describe it in a text query [105]. The text that appears alongside dataset images may not capture the semantics that the user requires [63], or have different wording than they expect [125], *e.g.* "car"/"motorcar"/"automobile". Users would generally not be aware of what terminology exists within the metadata [111], especially if the metadata contains expert knowledge on the subject. Neverthe-

less, regardless of the possible annotations, it should be possible to retrieve images even if there is little or no metadata at all [105].

These problems motivate us to rely on a retrieval system that can work using a visual query and by analysing the visual information [88] that exists within the image pixels (*i.e.* the visual 'content'). This is known as Content-Based Image Retrieval (CBIR).

Content-Based Image Retrieval

Humans are able to rapidly detect image similarity without accompanying text information [88], so it is intuitive for an automated process to perform a similar function. CBIR extends back to 1992 with the seminal work of Hirata and Kato (1992) [40] which used simple user sketches to retrieve database images according to their similarity to the sketch. This is a form of 'Query by Visual Example' [40, 133]. A user-chosen image or image region (from either inside the image dataset or externally) could also be used as a query [91, 111]. The system then returns to the user the image dataset (or a subset of the image dataset) in descending order of similarity to that query.

The increased computation performance and storage of modern computer systems allows for raw images to be used as queries, which is a more user-friendly method. CBIR potentially simplifies the process of dataset exploration by relying only on the presence of the database images and a query image (which may itself be a database image). However, the difference between the true meaning of the image content and any numeric features becomes a semantic gap [88, 91, 105, 111]. The high-level concepts may not be apparent in the low-level features or even the global features of the image.

CBIR operates in two main stages: the offline stage (Figure 2.1) and the online stage (Figure 2.2). The offline stage produces the descriptors for each image in the gallery, and constructs the database. The online stage should be computationally fast because it occurs whilst the user waits for the results. In this case, the query image is uploaded, its descriptor is produced, and is rapidly compared against the descriptors

in the database (this is in practice performed using fast matrix operations) before returning the relevant images to the user.

Production of useful numeric representations of images generally requires one high-dimensional vector (or 'descriptor') per image. This can be a descriptor using a global (whole image) feature [9] or the aggregation of local, low-level features [111, 112]. Images are represented within a gallery database as high-dimensional vectors [9, 48] where the similarity of two images is equivalent to the distance of their descriptors within the vector space. The similarity of a query vector I_q and a gallery image vector I_g is given by their cosine similarity:

$$sim(I_q, I_g) = \frac{I_q \cdot I_g}{||I_q||\ ||I_g||} \tag{2.1}$$

Calculating similarities between the query I_q and all n images within the gallery I_1 .. I_n can be performed as a single operation. Firstly, by constructing a matrix $I \in \mathbb{R}^{C \times n}$, where C is the vector representation length. The i^{th} column of I is the vector representation of the i^{th} gallery image. Similarity of query vector $q \in \mathbb{R}^{1 \times C}$ to each gallery image is performed with cosine similarity:

$$sim(q, I) = \frac{q \cdot I}{||q \cdot I|| \cdot ||q \cdot I||} \tag{2.2}$$

Ranking images as their similarity to the query is then done with a sorting operation.

Photographic Archives and Annotations

With widespread use of digital cameras and smartphones and the decreasing cost of digital storage it is very easy and cost-effective to produce and store photographs (Facebook reported the upload of 350 million images daily as early as 2013 [6]). Also occurring at a rapid rate is the digitisation of historical photographs and images. Libraries, museums, archives, and other cultural and historic institutions (both private and governmental) around the world are working to digitise vast collections of

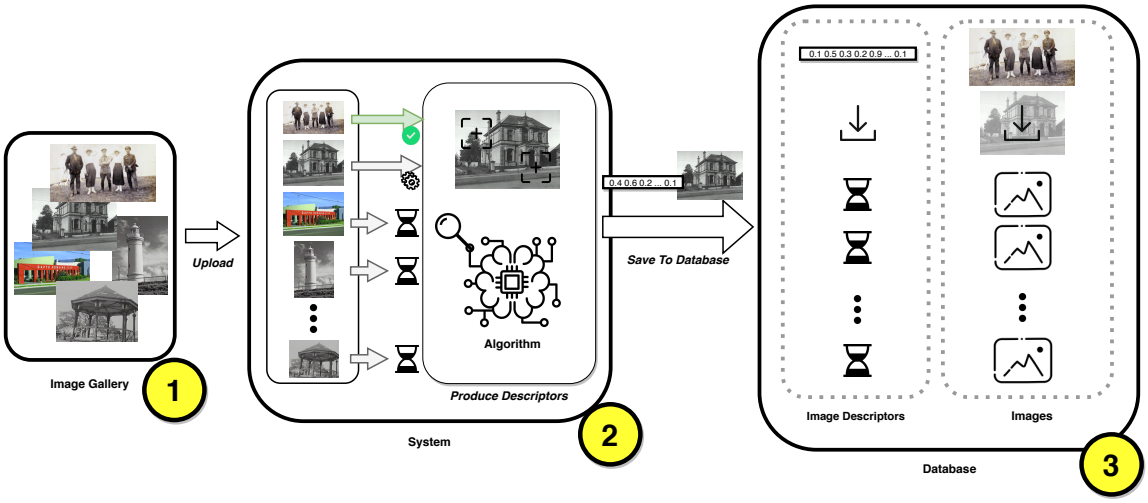

Figure 2.1: Basic process of Content-Based Image Retrieval in the 'offline' stage. (1) An image gallery is uploaded to the system. (2) Each image in the gallery is inspected by the algorithm and a numeric descriptor is produced. Each image and its associated descriptor are (3) stored in the database.

images and photography [3, 4, 5, 91, 109, 111, 125]. This ensures their preservation going forward and to minimise the cost of ongoing physical storage [112] of photographic material previously concealed (and protected) in filing cabinets, boxes, and storage rooms [14]. The digitisation process coupled with an accessible online retrieval system can allow everyday citizens to access and study images without expert knowledge, and to easily explore without a concrete objective or query. For digitised archival images and historical photographic collections, users may be interested in the exploration of local or national history and culture without a specific aim. Similarly, artists can use CBIR to find works in digital art collections [17].

However, most existing online image retrieval systems from libraries and museums require the input of keywords, date ranges, and other fields (which can be useful in their own right), but these require prior knowledge, a useful set of keywords, and correct spelling and choice of words by the user. Furthermore, it requires properly annotated descriptions and fields alongside the database images for them to be searchable (see Text-Based Image Retrieval).

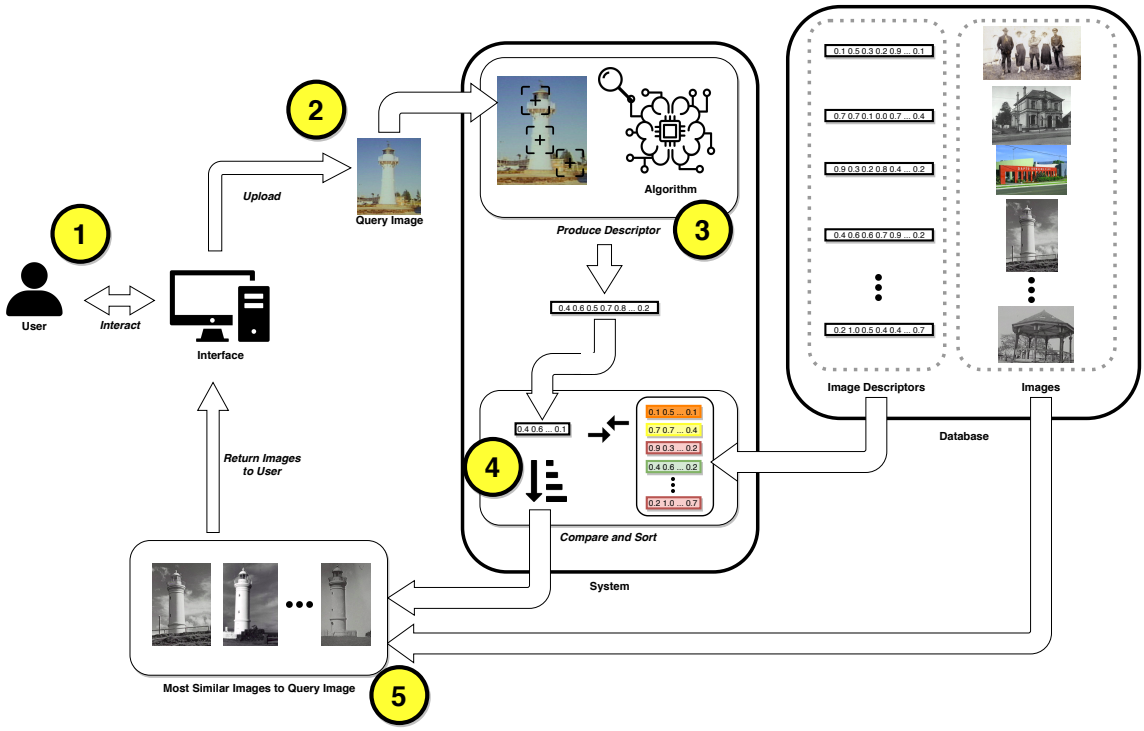

Figure 2.2: Basic process of Content-Based Image Retrieval in the 'online' stage. (1) The user interacts with the system and (2) uploads a query image. The query image is inspected by an algorithm (3) that produces a concise numeric descriptor representing the image contents. The query descriptor is compared to all the existing database image descriptors (4) and the database images are sorted by their descriptors' similarity to the query (5) before being returned to the interface.

Challenges of Archival Datasets

Archival, historical, and cultural image datasets are more challenging than traditional (public) image retrieval datasets [91]. Public datasets can contain highly semantic objects, and can vary in illumination, orientation, angle, and overall appearance [74]. Modern public datasets (such as Paris6k [90], Oxford5k [89], INRIA Holidays [46], and UKBench [79]) are high-quality colour photographs produced with digital cameras. The images have high resolutions and do not suffer from deterioration or image quality issues. By contrast, the images in archival sets are generally digital scans of older hardcopy photographs, artifacts, art, and are of varying indoor and outdoor locations [91]. These photographs have often suffered visual damage caused by physical deterioration over time from improper storage (like heat or water damage or edge disintegration), poorly-attempted repair, and mishandling. Digital scans may still include distracting features like colour separation charts, measurement rulers, borders, and page numbers, which need to be ignored by any IR algorithm. Furthermore, scanned images may be askew or off-centre, and are subject to changes in illumination, cropping, resizing, or may suffer from lossy compression during the digitisation process [14].

Since the images are of a historic nature, objects may also change in appearance through the decades (*e.g.* transportation) [20], while buildings can change in appearance (urban development and changes to facades). Objects and landmarks found within multiple images also have large changes in viewpoint and angle [111], and illumination and colour. Older photos may also be black and white (or sepia) while newer images may be in colour. Older images suffer from variations of colour and geometry [112].

Existing Retrieval Systems

There exist some early photographic archives, and some are available for public access and search. However, most rely on text-based retrieval only [40, 125].

The Eurovision St Andrews University Scotland photographic collection [19, 97] contains 28,133 photographs and postcards (some hand-drawn) from the library of

the University of St Andrews in Scotland. It consists of images of scenery, portraits, animals, and buildings. About 10% are in colour, and the rest in greyscale.

The Arquivo Público Mineiro collection [4] contains cultural photographs designated to be preserved by the Brazilan state of Minas Gerais, including landscapes, monuments, people, and daily living.

The British Museum Online Collection [3] contains over 1 million records that have images. However, the collection's search system relies on metadata and text-based searches, and there is no feature for searching using image as a query.

The National Archives of Australia (NAA) has digitised over 300k images onto their PhotoSearch [5] tool, which allows for users to search using keywords and a date criterion. However, the tool does not allow for direct visual querying using an image.

The Bodleian Ballads Search tool [2] from the University of Oxford is a working online image search system using query-by-example of either a whole or part image. It is rapid and uses SIFT features [67] in a bag-of-visual-words model. However, it contains only prints of words and simple, repeating images, with little geometric or semantic variation, such as those in the NAA images.

Finally, the Bibliothèque Nationale de France heritage image collection [1] is an online, searchable collection of images regarding French history and culture. It does not have a query-as-example feature, but allows for search by keywords.

Image Retrieval Using Archival Images

Image Retrieval using archival or cultural image collections was studied in the early- to mid-2000's [105, 111], with some recent interest [17, 20, 91, 109].

Archival images, due to their difficulty, are an excellent extension to the public image retrieval datasets, and represent a 'real' and practical instance of image retrieval. Furthermore, many archival image datasets come with existing metadata and other descriptive text annotations. While the problems with using text are mentioned earlier, the difficulty of archival images is impetus to employ available text annotations as part of the image retrieval pipeline. While the text in archival datasets does not

Figure 2.3: A set of random images from the NAA29k gallery

always contain full-form sentences, the user does not generally supply full-form sentences as their text search query [88]. Instead, users tend to use specific keywords to both find specific tags and to compensate for perceived weaknesses in the retrieval algorithm [88].

This thesis will utilise a developed archival image dataset named NAA29k, a collection of 28,912 images from the National Archives of Australia [5], as well as its accompanying text metadata.

National Archives of Australia - NAA29k Dataset

There is serious motivation to produce a useful benchmark archival dataset that can be used for Image Retrieval and have a groundtruth that allows retrieval models to be quantitatively measured for their content-based retrieval effectiveness. From the over 300k digitised images in the National Archives of Australia PhotoSearch database, 28,912 were selected to construct an archival dataset. The aim of this thesis is to use NAA29k alongside public benchmark datasets to investigate computer vision methods of Image Retrieval.

The images in NAA29k (Figure 2.3) represent a broad range of visual information that provides a complexity not found in public baseline Image Retrieval

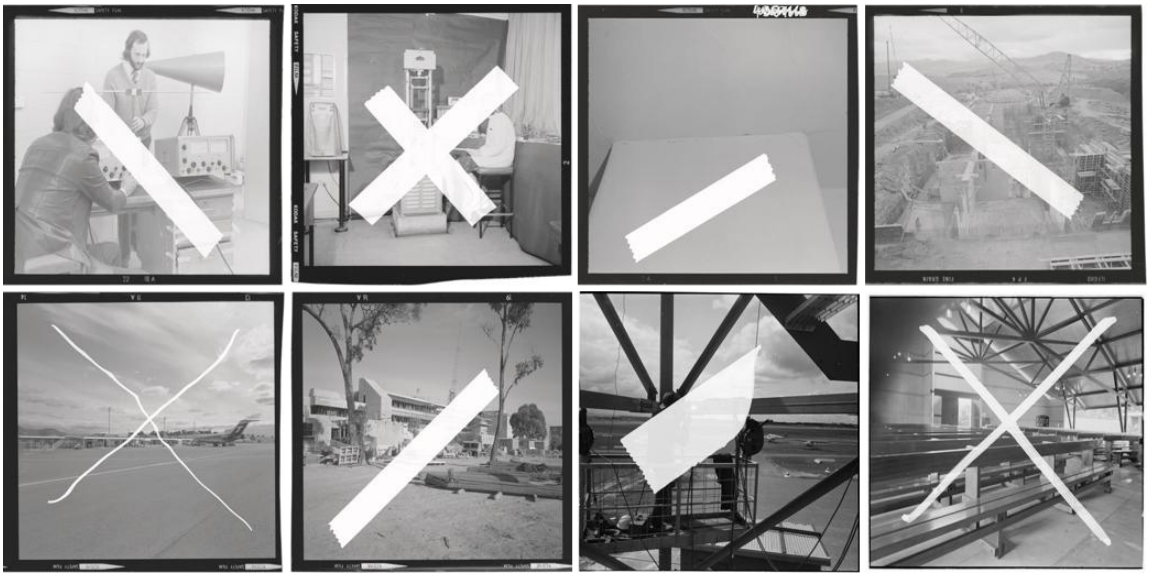

Figure 2.4: Examples of damaged images in NAA29k that have been taped over, or otherwise marked, which occludes portions of the image

datasets. The gallery is broad in scope and content, and includes scenery and nature, streetscapes, national landmarks, buildings in regional cities, building models, and various indoor photos. There are also images of construction works, portraits and group photographs, sketches and drawings, political events, and more. The gallery captures significant events, places, and people in Australian history and culture. The images range between the years 1833 and 2002. We counted 3,785 either colour or sepia-tone images, and 25,127 greyscale images.

The original source images were of varying sizes, and we resized them to be 256 pixels on the larger side, then took a 256×256 center crop. No borders or other features were deliberately altered. Therefore there are still significant visual problems that would make this dataset more difficult than high-quality public datasets. Many images contain noticeable 'noise', including grainy appearances, fade, and damage (Figure 2.5). Many damaged (and repaired) images are obscured by tape and markers (Figure 2.4).

Many images also feature other items such as charts and book pages (Figure 2.5), or are otherwise adversely affected by age and deterioration. These features may

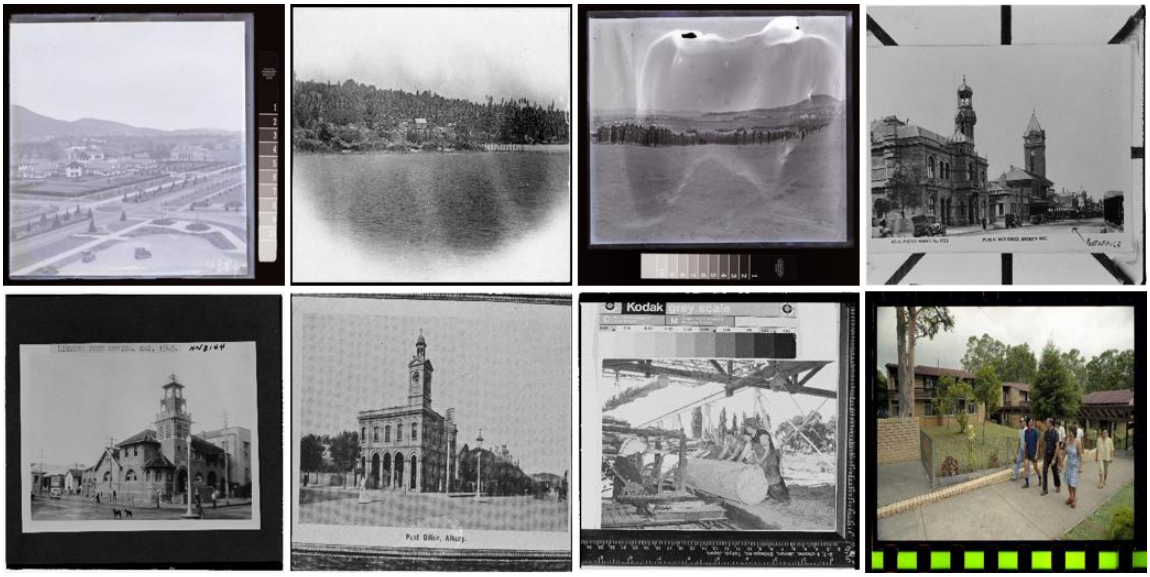

Figure 2.5: Examples of archival images in NAA29k that have distracting features like heat damage, fading, colour separation charts, book pages, rulers, and excessive borders

distract from the salient features in the photographs.

Each image in NAA29k is paired with metadata that includes a manually written description of the image. The text corpus is varied, with rare words appearing only in one image each, whilst other words occur across several thousand images (Figure 2.6). While image annotations vary in length and detail, most describe high level semantics, such as the names of buildings, notable people, or an event taking place. These annotated descriptions can suffer from contextual limitations, but can also provide an extra clue to complement the visual information. The descriptions are susceptible to annotator knowledge, subjectivity, and human error. For example, the phrase 'Prime Minister' occurs in 345 image annotations, but sometimes is not accompanied with the leader's name, and can occur for visiting prime ministers of different nations. Another annotation of a mining photo contains the words 'Kambala Lake Lefroy', but *Kambalda* is the correct spelling of the mining town in question. Despite such problems, the annotations can provide well-detailed descriptions, with one boasting a 155-word description of the background of a hospital and its founding by an immigrant couple. Others have shorter descriptions (198 images contain only

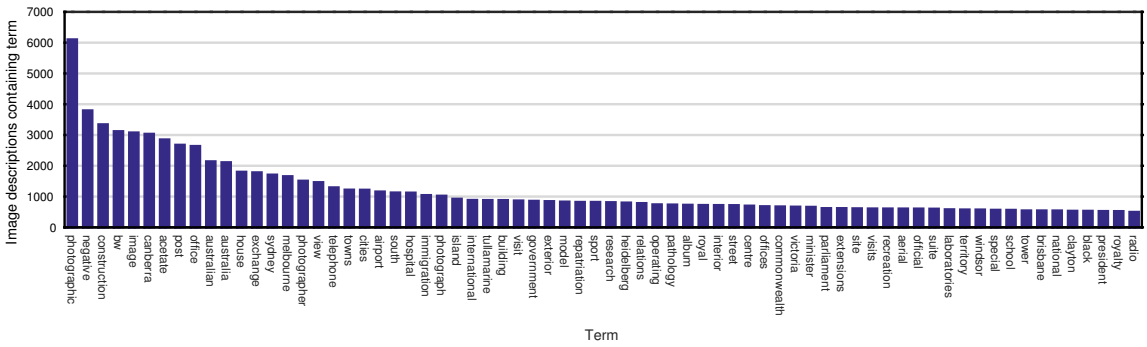

Figure 2.6: The 70 most common terms in the NAA29k metadata, excluding common stopwords. The words 'photographic', 'negative', and 'bw' (black & white) were included in annotations to indicate the image source type

a single word), but still provide extra context to the image.

Without considering special characters and letter case, there are 12,039 unique descriptions across the 28,912 images. This also does not consider combining word variants (australia / australian / angloaustralian, office / offices, display / displaying / displays / displayed, *etc.*). This occurs because many groups of images (say, from the same event) are labelled with the same description. This is a useful aid in linking images that have no obvious visual similarities or landmarks. Some images of the largest group are shown in Figure 2.7. Metadata can be very useful as a guide in archival images where additional context and background would be useful to archivers and researchers with a specific information need. The additional textual information can be used as a 'guide' to complement visual image retrieval.

To produce the groundtruth, a series of images were selected to be queries that contained some visually-similar counterparts in the gallery. For example, buildings, scenes, and people. Two groundtruths were created: one with 1,137 queries, and another with a further refined 100 queries. The 100 query set has no fewer than 6 and no more than 18 positives. These will be referred to specifically as NAA29k_{100} and NAA29k_{1137}.

Figure 2.7: Some of the 253 images in NAA29k with the description 'Tullamarine Airport special extensions' (when special characters are omitted)

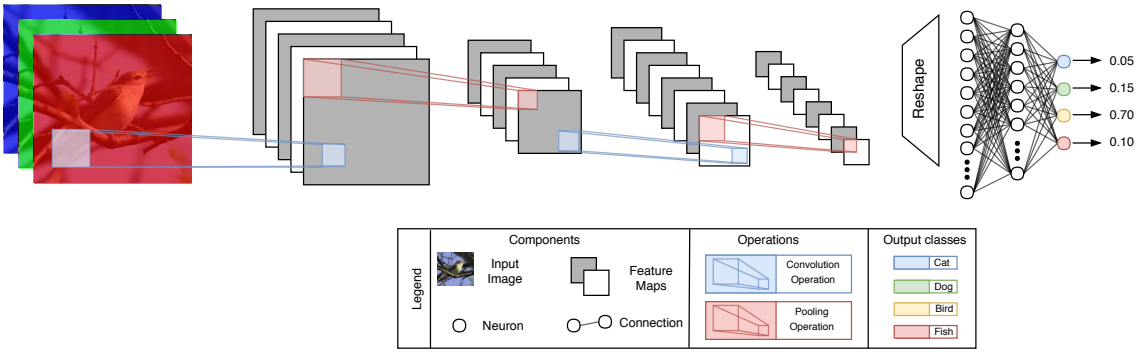

Figure 2.8: Simple example of a Convolutional Neural Network trained to classify images as either cat, dog, bird, or fish. Alternating convolution and pooling operations produce feature maps at each layer, followed by a fully-connected stage and a softmax output. The bird image is fed into the CNN and a four-element output is produced.

Convolutional Neural Networks for Image Retrieval

Image Retrieval relies upon a similarity mechanism for numerically ranking reference images to a query image. Early on, the pixel-level information of images was harnessed [105, 111], including colour [34, 43, 83, 99, 113, 126], shape [14, 43, 51, 100], and texture [52, 73, 127]. Later, hand-crafted features including SIFT [67], SURF [11], as well as Grayvalue Invariant Features [100] produced good results. These hand-crafted features are invariant to some minor geometric changes. Arguably the most popular feature, SIFT, can be used for object detection and image retrieval even under rotation and changes in scale. In effect, these features are densely extracted, representing salient points in the image. Despite their power, they suffer from high dimensionality, are computationally heavy to generate, and is (along with SURF) patented, requiring a paid licence for commercial use.

More recently, however, Convolutional Neural Networks (CNNs) [58] have been utilised as a means to extract dense and powerful image features. The ability to boost computational performance with GPUs (Graphics Processing Units), the availability of pre-trained 'off-the-shelf' networks, and open-source machine learning frameworks make this an attractive tool for image retrieval. CNNs have great utility because they maintain important spatial information from the input image. While CNNs

have been used for many image tasks such as classification [32, 37, 42, 61] and object localisation [102, 135], the trained networks have shown good performance on the task of image retrieval [9, 10, 48, 74]. Off-the-shelf networks, although trained for the image classification task, have shown incredible performance in the image retrieval task, whereby extracted features from the FC layer [9, 10] can be exploited to produce discriminative and powerful image descriptors. The main question is how to effectively pool the dense image features produced by the CNN [9, 35, 37, 48, 76]. Some parts of the image features are more informative than others [130], motivating different pooling techniques, and even inspiring newer, efficient, CNN structures with less redundancy [39].

Visual Grounding

Visual Grounding, or localisation, is a task to localise a text phrase onto an image by selecting the position of the visual object the text describes. Model training can include fully-supervised methods using annotation bounding boxes [116], however Weakly-supervised Object Localisation (WSOL) is a major focus [80] as it does not require expensive bounding box annotations during training. There is practical use to this visual task, including application in the medical field to use for medical instrument detection during surgery [114].

Using a set of known classes and a small neural network, a Class Activation Map can be used to perform localisation [122, 135], but does not rely on specific text information, only class labels. Localisation of object parts and more coverage of object areas can be performed using multiple object detectors [28] or combinations of activation maps [122]. The text encoding can be performed using an off-the-shelf pre-trained text embedding [70] or the embedding used as input into a trainable module [53].

Deep learning models use encoder-decoder frameworks [44] or the embedding of multiple features into a common space [29] trained on publicly-accessible groundtruth datasets where both image and text are known. Such models can also expand cover-

age of objects with multiple detectors that perform partial erasure on one detection to allow a second detector to detect less discriminative parts of the object [132].

Layout of this Thesis and Novel Contributions

The following chapter provides a review of the literature in the space of the history of image retrieval, and the use of deep features from convolutional neural networks in modern retrieval methods. Then the review explores region pooling, and discusses the usefulness of selective pooling. Then it moves to the task of visual grounding, and how text can be exploited for the task, and the possibility of attention provided by text annotations.

In "CNN-Based Image Retrieval" we explore the image retrieval task, especially on an archival dataset of digitised images from the National Archives of Australia. We explore the effectiveness of using various models and pooling techniques and visualise retrieval on the archival set. Delving deeper into a weighted pooling method called SPoC [9], it is identified that it is sub-optimal for parameter choice, and we propose a simple improvement that can boost retrieval performance for specific datasets.

In "Visual Grounding Utilising Word2Vec Semantic Structure" the task of Visual Grounding, or text localisation, is covered where we design a localisation model that is trained end-to-end without the need of a specific trainable text module. Using a pre-trained Word2Vec text embedding, we harness its semantic structure to perform localisation on unseen words and demonstrate how it performs on publicly-available localisation dataset. Further contribution is proposed modifications to the model's object localisation module, including the use of erasure technique to coerce improved learning of visual objects.

In "Text Guided Archival Image Retrieval using Localisation Model" we demonstrate the use of our trained localisation module on the archival dataset, by exploiting the available text annotations and producing text-guided heatmaps for improved pooling. Our novel contribution is to weigh visual features on the archival dataset with our trained localisation model. We highlight that the weakness of our approach

utilising text relies on quality metadata with rich visual descriptions, and that we propose a simple solution to balance the influence of our text-guided module. We demonstrate that our trained model can weigh features to produce a modest improvement over the baseline methods, and show that our method can be used in future image retrieval work involving galleries with accompanying text metadata.

Finally in "Conclusion" we close out our discussion and highlight our findings.

Chapter 3

Literature Review

Introduction

Producing image descriptors for the task of Image Retrieval (IR) has recently shifted from using hand-crafted local image features to the activation features from a trained Convolutional Neural Network (CNN). While the outputs fully-connected layers of CNNs can be used to produce effective global image descriptors, the current approach is to use output activations of convolutional layers [9, 10, 41]. Convolutional layer activation outputs can be considered as 3-dimensional tensors containing neuronal activations of visual features, while also maintaining useful spatial information from the input image. However, the question of *how* to utilise the activation tensor to generate a concise descriptor is an active research question, leading to numerous techniques and approaches for pooling and aggregation. Simple global pooling has been bested by more complex strategies of multi-scale pooling, region masking, and region-based features. Although these approaches are broadly hand-crafted or unsupervised, the weighted pooling approach has been critical for improved IR performance.

Region-based pooling is generally naïve and does not take into consideration the image contents as a guide. Similarly, the tensor weighting has been handcrafted. Attentive methods for pooling harness the power of automated learning to attribute more suitable weighting to activation tensors. The self-contained module approach

is the newest notion with modules that can be trained end-to-end with the CNN.

This chapter critiques the techniques and strategies for the extraction and use of image features for IR. The lack of literature regarding the exploitation of available text information with archival images as an attentive guide motivates research into this space. The anticipation is that such a method can further improve the performance of the image retrieval task, especially with more difficult archival image datasets that also have text descriptions.

CBIR: The Beginning

Query by Visual Example (*i.e.* Content-Based Image Retrieval, or CBIR) was first proposed in the context of art and cultural collections [40], whereby a user (recalling an artwork from memory but not its name or artist) could draw a rough sketch to retrieve it. Algorithm-generated sketches are made in the off-line stage using the maximum gradient of intensity changes in four directions, producing small (64x64 pixel) binary sketches of all dataset images. By sliding a correlation filter over the sketches, the user's sketch is compared to the database sketches to return the closest images. The lack of computation power at the time necessitated the use of simplified binary sketches for comparison, placing a burden on the user to produce their own query. However, it did introduce the idea of query by example, especially as a novel means of providing a query. Modern computing power allows whole, raw, images to be used as queries (the visual example). Features of the image can be extracted at many locations on the image, which can be aggregated to produce a descriptor.

Hand-crafted Features

Pixel-level features of images can be exploited to produce image descriptors using colour, gradient, and shape information. Basic colour histograms [107] are counts of the occurrences of colours in a finite number of 'bins', thereby capturing the colour distribution of the image. While this can be extended further to more complex colour

moments and histogram variants [99, 113], it is inherently an insubstantial method for complex images. The weakness in using colour comes from its lack of discriminative power - many different objects and scenes share similar colour distributions, and it is ineffective on greyscale images. The more descriptive gradients of intensity changes can instead be utilised without the need for colour, and can still be binned into histograms [23]. This captures the information in basic shapes. While this takes the spatial information and simple, local features into account, it still does not capture semantic objects and high-level features. Different objects can contain similar low-level visual features despite having semantic (*i.e.* high-level) dissimilarities.

SIFT features [67] can outperform colour features [112] due to their rich high-dimensional descriptors at visual keypoints, and their invariance to geometric changes including scale and orientation [67]. SIFT features are powerful for finding the same object or feature even at different rotations, and has thus been used in the image retrieval task [9, 47], while its detector feature is useful for finding highly semantic parts in images [41]. Using SIFT features for CBIR in an archival/historical image dataset [112] showed improved performance over simpler colour histograms [84, 127]. Further, their method maintains spatial positioning for keypoints, in contrast to the global colour histogram that removes all spatial information when making a descriptor [60, 112]. However, this can lead to rapid increases in computation time as the size of the dataset increases. The more general approach is to aggregate the 128-dimensional SIFT features into a single vector using Fisher Vector [98], the simpler Vector of Locally Aggregated Descriptors [8, 47], or using the Bag of Words model. Production of long descriptors (tens of thousands of dimensions) [8, 47, 98] introduces problems of storage requirements of online search speed unless dimension reduction is used. Nevertheless, the resulting descriptors are more powerful than colour descriptors for the IR task.

SIFT [67] and other hand-crafted features like SURF [11] are not robust to large changes in viewpoint (*e.g.* looking at a building from two different sides). Thus for complex images in image retrieval the hand-crafted features do not capture their high-level semantics that remain after viewpoint change [22, 41, 136]. Images that have the same objects or contents can be visually diverse, adding unrestricted complexity

to the retrieval task [105]. The need for a more powerful descriptor is required to be robust to changes caused by rotation and scale.

Convolutional Neural Network Features

Features extracted from a Convolutional Neural Network (CNN) can be more powerful than historically hand-crafted features. Furthermore, the use of an automatically pretrained (or "off-the-shelf") CNN removes the need to manually hand-craft a local descriptor at all. CNNs are networks of trainable weights in stackable blocks designed to mimic the human vision system [78, 130]. The filters of earlier convolutional layers correspond to the detection of simple patterns and shapes [78], while later layers detect more semantic or object-like features [74, 134]. This contrasts with the SIFT features [67] that correspond to medium-level features, and low-level colour and gradient features [99, 107]. Comparing structurally-similar networks trained on different data, the earlier layers 'look' for the same type of simple features, such as blobs, colours, and edges [134]. Later layers in a network correspond to higher semantics, such as objects and scenes. CNNs are usually constructed with a series of convolutional layers (separated by non-linear layers such as the Rectified Linear Unit), and finally by a Fully-Connected (FC) module consisting of several fully-connected layers [58]. This gives many opportunities to extract features, including most simply at the final FC module.

Generally, it is desirable to represent each image as a single vector [9, 10, 48, 91]. The output of the penultimate FC layer can act as a compact global feature vector to represent the image, known as a descriptor [9, 10, 91]. Using FC descriptors can outperform simpler, hand-crafted features while being of lower dimensionality [10, 20], lessening the need for a post-processing dimension-reduction stage. For object classification in a dataset of digitised paintings, FC features extracted from a CNN are superior in performance (and in storage requirements) to very-high-dimensional Fisher Vectors with SIFT features [20]. However, this method was limited to an implementation of 12 object classes. Models trained for a large number of classes

show that indeed the FC descriptors can be used as useful discriminative vectors, especially with network fine-tuning to improve retrieval performance [10].

Use of the FC module has two main limitations. Firstly, it always takes in the full image content, including any noisy parts without considering the relative relevance or significance of the parts or regions [35, 48]. This may not properly describe or encode the smaller objects [91], properties, or layout of parts [35, 72]. Secondly, as the FC module was designed to handle a finite set of output classes, there will be a mismatch between the domains of the trained dataset and the test dataset, whereby object and scenes may differ. Furthermore, the use of the FC module requires input images to be of a pre-determined size. This forces the input images to be cropped or reshaped, while also not taking into account any specific spatial information. However, it is possible to explicitly save spatial information using the FC layer outputs.

Re-feeding separate image regions into the network can produce separate FC outputs that contain location-specific information, but this has drawbacks in computation performance and domain mismatch. This technique involves dividing input images into spatial regions and feeding them separately into a pre-trained CNN to extract their FC outputs [35]. The features were pooled at different scales using VLAD [47] and then concatenated to form one descriptor for the image. However, this suffers from a domain mismatch problem [130] whereby the pre-trained network trained on whole, singular images is used for small image regions and possible non-objects, reducing the discriminative power of the region descriptors. Furthermore, larger numbers of locations and scales drives up the dimensionality of the VLAD descriptors (and thus the final descriptor). Lastly, and most importantly, each and every desired region must be resized to the correct size and fed as input into the CNN, which has substantial computational cost. To solve these problems, the FC module can actually be ignored, and the activation of the previous convolutional layer used instead.

Convolutional Layer Activation Pooling

The convolution layer outputs contain richer information about the input image than an FC descriptor, and methods that utilise these activations can pool spatial information into the final image descriptor. There is a conceptual parallel between CNNs and human vision, whereby CNN deep features are the most powerful current means that replicate the human concept of image similarity [130].

The output activation at any convolutional layer is a 3-D tensor $\boldsymbol{X} \in \mathbb{R}^{H \times W \times C}$. The tensor can be considered in two ways: (1) as C 2-dimensional channels (also called 'feature maps') each of size $H \times W$, or (2) as $D = H \times W$ number of deep features, each of length C, one at each of D spatial locations. The activation should be taken after the ReLU unit to remove negative values [82]. Spatial information in the activation tensor highly corresponds to the features in the original image that lead to those activations [87, 121, 134]. Removal of the most salient features in the input image correspond qualitatively to the largest changes to the activation tensor [134]. Spatial feature removal can also be used to enhance feature detection and object localisation [12]. The 3-D activation tensor at a convolutional layer should be pooled to form a compact and useful image descriptor, especially in later layers that contain more semantic information [75, 128, 134].

Collapsing each channel to a scalar will produce a vector of length C. The simplest pooling techniques are SumPool [9] and MaxPool [95]. The summation method, SumPool [9], accumulates all activations across the spatial area, but does not have the small-object localisation power as in MaxPool [95], in which smaller but highly-semantic objects can highly activate, including infrequent features [76]. Postprocessing the SumPool feature [9] with a simple square-root operation can further increase retrieval performance [64]. Further, a hybrid approach using multiple descriptors (AveragePool and MaxPool) can increase retrieval performance despite doubling the storage requirements [75].

Weighting techniques [9, 48, 76] can weigh convolutional feature activations to focus on the most discriminative deep features, or can weigh channels as well [48]. Weighted regions of interest are areas that should intuitively present the most infor-

mation about the original image's contents [112]. Extending SumPool with a center weight gives more importance to any center-focused objects to produce their SPoC (Sum-Pooled Convolutional Features) descriptor [9]. However, this is performed in an unsupervised manner using a hand-picked gaussian weighting function over the activation to increase weight at the center. This relies on the broad assumption that the center deep features (and thus the center of the original image) contains the salient information. While this may be the case for some curated image benchmark datasets such as ImageNet [24], in difficult retrieval images the important features can occur at any position or scale within the image, including nearer the edges. This is avoided in the spatial- *and* channel-weighted CroW (Cross-Dimensional Weighting) feature [48]. Since the channels of the convolutional activation correspond to learned semantic features or objects [41, 128], the tensor could be better weighted using the channel information as well. The CroW feature [48] is produced using both channel weights and spatial weights. Channel weighting is determined by overall channel sparsity, while spatial weights are built using an accumulation of the feature maps. Although this produces a more powerful descriptor, much like SPoC it is a hand-made method that does not use explicit learning. However, it does highlight that the spatial and channel information in the activation can be directly harnessed. A combination of multi-layer activations, feature masking, and region-based selection can enhance CroW features, with improved performance on the public CBIR datasets [87]. It is region-based pooling that can leverage the spatial information retained in convolution activation tensors.

Region-based Pooling

Region-based descriptors can be produced from separate feature extraction of multiple spatial regions of an image. Extracting FC descriptors at different image regions and scales in a pyramid-like manner, and aggregating with VLAD [8] , can produce useful global descriptors [35]. This approach has the problem of domain mismatch where the regions are not in the same domain as the training data [64, 130]. Although

they improve upon the basic VLAD implementation by treating each pyramid layer separately with separate visual codebooks, this still requires the computationally-expensive process of feeding each region into the CNN separately to extract each region feature. Since the convolution activation features highly correspond to their respective positions in the input image [121, 134], a single convolutional feature activation can be taken at a chosen convolutional layer, and the desired region features extracted from the activation at different regions and scales. This jointly avoids the problems of (1) strict input image size, and (2) multiple expensive feeds through the CNN. It can also allow for the discovery of objects within the image [20] and specific regional or object parts [91].

The choice of locations and scales for regional feature extraction is often done in a naïve manner depending only on the desired size of objects [8], and the rectangular window regions may not nicely align with objects or features within the input image [102], or not properly encompass a salient region of interest that is of arbitrary shape and size. To include spatial information in spatially-ignorant VLAD pooling, SIFT features can be extracted at multiple spatial windows and then pooled [17]. However, this performance of this method was not compared to plain VLAD, and instead uses a weighted sum of the multiple methods [17]. The Regional Maximum Activation of Convolutions (R-MAC) descriptor [110] takes MaxPool descriptors at different scales and different sizes across the activation tensor rather than separate feeds of the original image. This means the image is only fed through the CNN once, and all post-processing done on the 3-D CNN activation [64, 110]. While the best choice of the number of layers [110] in the pyramid was tested empirically ($L = 4$), the size of the regions and their overlap (stride) was hand-crafted. Similarly, the SIFT-based MultiVLAD scheme [8] uses a pyramid structure of VLAD descriptors at hand-picked scales and region sizes. Their regions are non-overlapping, potentially 'cutting' potential objects between adjacent regions, as opposed to the partially overlapping regions in [110]. However, the SIFT-based MultiVLAD algorithm is still outperformed by deep-feature-based R-MAC in terms of both retrieval performance and storage requirements (32768-D vectors vs. 512-D vectors). For the better-performing R-MAC algorithm [110] the region vectors are naïvely accumulated to form a final descrip-

tor, whether or not the regions contain features or noise. Intuitively the important foreground features may be spatially small, and its activation can be suppressed by significant distracting background [119]. Similarly, extracting simple SIFT and edge features in regions can produce descriptors from histograms [57]. These methods rely on naïve region segmentation to extract features at different scales and locations regardless of the presence of useful information [35, 110]. It is therefore more useful if regions or even specific parts of activations receive proper weighting depending on their actual 'objective-ness' or saliency in an image-dependent manner.

Selective Pooling and Saliency

The individual feature maps (*i.e.* channels) in convolutional activation tensors are essentially activation responses by a set of highly-semantic feature (or part) detectors [37, 41, 64, 74, 128]. The 'parts' may not correspond directly to human-annotated parts, and may instead be high-level details learned by the network [134]. Thus, two images with similar features or objects will have high activations on common feature detectors, and low activations on detectors that find features not in the image [37]. However, frequently-appearing spatial regions in the pixel space across the image or the entire gallery (*e.g.* sky and clouds) may not be useful to include in the descriptor. In contrast, rare objects or patches may be highly discriminative [74, 76, 77], and are therefore more salient than others [2, 20, 91, 102, 105, 118]. The most relevant points could be maintained to produce useful image descriptions [105], and such regions could be appropriately weighted in the pooling process. Producing Bag of Visual Word descriptors is possible with SIFT features, but using a keypoint detector kind find the appropriate places for SIFT feature extraction [63]. While this approach succeeds in avoiding irrelevant visual words from entering the visual vocabulary, the methodology uses only simple gradient-based methods for keypoint detection. These keypoint detectors such as Laplacian of Gaussian and Difference of Gaussian [36] and the SIFT detector [66] indeed find more suitable points of interest, but in the domain of CNNs this is equivalent to the learned early layers for use as feature detection. This

idea is partially applied in the cross-convolutional layer pooling strategy [64], which uses one layer's activation as a guide to weigh deep features in the *previous* layer for a weighted SumPool operation in order to use the most appropriate spatial features. However, this pooling approach concatenates the exhaustive multiplication of the feature maps from two layers, thereby producing a final vector of substantially larger size (*e.g.* $256 \times 256 = 65536$). To alleviate this, they train a new convolutional layer with fewer feature maps and use post-processing PCA dimension reduction. Keeping only the top k activated feature maps for descriptor production can increase performance and reduce noise from irrelevant feature detectors [64]. They experimentally show that the removal of some lower-activating channels increases performance. Alternatively, a bottom-up approach can first generate spatial image regions and then decide whether they contain objects by a threshold level of 'objectness' before classifying parts of objects [119]. This approach requires simple labelled groundtruth to train the objectiveness classifier, but is then limited to seen objects rather than overall objectiveness.

Finding generic salient parts can be performed with a spatial binary indicator mask to exclude deep features which are not discriminative, such as the sky in outdoor images [41]. Higher deep feature activations at particular locations correspond to objects or features. While their method is a simple approach for a baseline of selective pooling, they do not take the opportunity to use the l_2-norm values of the locations as an indicator of objectiveness [9], which can provide another baseline. Their methods are unsupervised and are fundamentally hand-crafted, much like the pre-CNN keypoint detectors [63]. While they remove non-discriminative features before pooling, their masking is just a pre-aggregation step [41]. All remaining features are just combined using existing techniques. However, it was still outperformed by the R-MAC approach [110] for the same descriptor dimensionality, but enjoyed a slight increase in performance on the scene-heavy Holidays [46] dataset and the UkBench [79] dataset for larger dimensionalities. Importantly it produced improved results over the computationally-heavy MOP-CNN [35].

Selective masking can occur either in the convolutional activation or at the input image. Instead of masking out irrelevant features in a weighted or binary fashion, an

alternative method is to focus on image objects that are salient and then MaxPool the representations of those regions in which the objects are located [74]. They exhaustively feed the cropped image region of *each object proposal* through a pretrained CNN. Although the strategy is much like MOP-CNN [35], only object proposals are used, rather than all locations at different scales. However, this is still computationally expensive, and it requires up to 1000 proposals (and CNN feeds) per image to reach peak retrieval performance, compared to using proposals in the CNN activation directly, which would reduce computation cost.

Masked Attention

When looking at an image, humans do not 'take in' the entire image [56, 118]. Scenes can be recognised by a small selection of features or objects without needing to take in the entire image [134]. Image queries can be reduced to correctly classify scenes when significant amount of extraneous features are removed [134]. The subconscious approach is to fixate upon salient features while ignoring irrelevant areas [56, 131] using decisions about what features are important or relevant to look at [45]. This can help avoid irrelevant features while focusing the attention on relevant features [121, 131]. The methods examined so far use a combination of region proposals or deep features with types of spatial or channel weighting. However, the weighting is generally hand-picked based on intuition, or require certain image parameters.

A controller module is used to learn and dictate which areas are best to fixate [56]. This is equivalent to a weighting in the spatial dimension in the pixel-space, and can correspond to those areas that have semantic objects or rare visual features. By focusing on actual regions that require attention [77, 102] the uninformative parts can be ignored [105, 131]. Using classification to produce spatial bounding boxes that conform to more discriminative parts of images can improve detection performance [102]. [37] train a new module *between* the convolutional stage and the FC stage in a CNN that spatially pools the activation regions to produce a pooled vector of the pyramid layers. The proportional sizes of the 'bins' in the pyramid layers allows

the input image to be accepted at any arbitrary size while still retaining an FC layer for classification. However, the authors only train a CNN from scratch and did not try using a pre-trained network and placing their module and a new FC module after it for training. This could balance the power of the new module while maintaining the less expensive use of a pre-trained network. Likewise [93] attempts both learning a shared vector of weights for weighting the channels, and a single weight for all channels in a power-scale manner, beating the IR performance of the R-MAC strategy. [71] train a separate CNN in a supervised manner to find the most salient features to adaptively weigh those important areas for the production of image descriptors. Also using the strategy of new CNN modules, the Squeeze and Excitation network [42] produces a trainable block that accepts a convolutional activation tensor as input and learns the appropriate channel weights in a small FC module (the excitation) in an end-to-end manner. This method is simple but powerful because the presence of activations on particular layers (thus the presence of objects) has non-linear relationships to activations in other layers [42, 134]. Accordingly, the presence of some objects corresponds strongly to the presence of other objects; *e.g.* boats are found on water, cars have wheels, and buildings have windows. Likewise, [131] learn channel attention weights *and* spatial attention weights. The channel attention is learned by a squeeze operation as in [42] and a 1×1 convolution layer, while the spatial attention is derived using learnable weights. More recently, an attention module to fit into the architecture of CNNs can learns in an end-to end manner [118]. They experimented on object detection and classification with improvement over the performance of the SE module. However, like their examinations did not specifically involve the image retrieval task [42, 118]. They further did not try using *only* spatial attention without channel attention in their experiments.

These newer methods have a focus on specifically learning the weights, but rely only on the image content (pixel content) as a guide. Since archival images can have associated descriptive text metadata, despite incompleteness or mistakes [109], it may provide for further enhancement during the spatial and channel learning.

Towards Visual Grounding and Text-Guided Attention

Humans desire specific visual features within an image to be useful and relevant to their query, related to the user's intention [111]. The user may be interested in a specific object in the foreground or in the background [88]. Therefore, an automated algorithm should either detect and ignore such irrelevant features [131], or take further user intention to focus away from those features and towards relevant features. User intention can be further refining using text-based inputs.

Text annotations can be utilised as a training signal to guide attention on convolutional activations. This has the benefit of minimising multiple feeds per image, and can leverage the high-level semantics in the text descriptions. [116] trained an attention network using existing class tag groundtruths for each image in the classification task. The learned attention detector then assigns spatial weights to the convolution activation of an image according to a given class tag. The authors did not extend their analysis to common public datasets (*e.g.* ImageNet [24]), nor to the image retrieval task. Since text words and sentences can be embedded as a single vector [116] and treated as classes, text can be used directly in model learning.

The Class Activation Mapping (CAM) [135] performs localisation of a finite number of classes by learning linear combinations of pooled feature maps from a CNN backbone. Their method relies on image-level labels and implicitly uses a linear combination of filters as object detectors. However, the learned filter combinations only focus on the most discriminatory parts of each object class, such as the head of an animal. To capture more of the object's entirety [122] simply fuse a series of CAMs by considering combinations of high- and low-discriminative activations.

For many visual tasks it is desired to learn more complementary object features, and the two usual methods are erasure, and a multi-kernel approach. In the erasure approach, the intuition is to drop out some input information to allow the network to better generalise the object and focus on its entirety. [104] do this by performing naïve data augmentation on the input image by removing random squares before

model input. Instead of random erasing, [117] use a series of networks to first discover discriminative object areas, then subsequently erase these 'interesting' areas from the input image before the next pass. This forces each network block in the series to localise complementary areas of the object, but suffers from the requirement of multiple forward passes. To avoid the need for multiple forward passes, [132] use a Fully Convolutional Network (FCN) and use its output as the input to two complementary networks. They perform erasure on the FCN's output feature maps to avoid re-feeding the image each time. Furthermore, their method of CAM using a convolutional kernel before the pooling layer produces class-wise feature detectors that can perform detection spatially during the forward pass. The goal of the multi-kernel approach is to use class-wise feature detectors. [28] uses a set number of feature maps to act as object part detectors. Each part detector implicitly learns different visual features of the object and the detectors are combined to form the object detector. [114] use this method in their practical application of medical tool detection. However, there is no direct or explicit guidance that trains what parts each detector should detect. In their network structure for person re-identification, [124] train a dual-path (text and image) model using a set of parallel network blocks that is equivalent to part-detecting kernels. Instead of a feedback that performs erasure on the input [117, 132], their model is trained to avoid detection overlap using a loss function that penalises the networks for detecting the same features. However, the Re-id method [124] relies on the fact that for that task the objects (pedestrians) are already localised and cropped. This still draws us to the hypothesis that a set of parallel object detection modules should act towards finding different object information and could boost visual grounding performance.

The use of cropped images, or images with annotated bounding boxes, is a form of supervised object localisation. Annotations are expensive, and being able to train for localisation on large datasets would require reduced supervision. [80] perform Weakly-Supervised Object Localisation (WSOL) by using a CNN followed by a max-pooling layer to perform class detection as a form of localisation. As opposed to [135], a set of convolution filters act as explicit linear combinations of the previous layer for localisation. Their trained model [80] also predicts object locations in cluttered

scenes without the need for location knowledge or bounding boxes during training. [28] learn object sub-parts with multiple convolution filters to localise different object parts before pooling to form a single object detector. Training is also performed using image-level labels without bounding box annotations, and they use Weldon pooling [65] to boost performance with a combination of max- and min- pooling with a non-learnable parameter that is empirically chosen. [114] perform object grounding similar to [28] as an application in the medical domain by detecting 7 surgical instruments in surgical videos using multiple convolution filters followed by the non-differentiable spatial Weldon pooling [65].

In the Visual Grounding task phrases can be grounded as whole text or by treating nouns as object classes. [44] use an encoder to jointly encode visual and text encodings with a non-linear operation to produce heatmaps, followed by a decoder that predicts concepts. Their class-based approach considers treating nouns as abstract visual classes (they use the term 'concepts'), and produce training batches according to concept presence, rather than using phrases at random. In contrast, [53] train a dual-path model using a Long-Short-Term Memory to encode full text sentences, and a separate CNN to encode images, into the same high-dimensional semantic space for the purpose of multi-modal retrieval. Similarly, [29] train a dual-path model to perform multi-modal retrieval but using a Simple Recurrent Unit (SRU) as the text encoder instead of an LSTM, due to improved computational performance. Despite training directly for multi-modal retrieval, the convolution unit inside the visual path in [29] can be re-purposed to perform visual grounding due to being implicitly trained to detect visual concepts, as in the case of [80, 114]. [29] select and aggregate specific visual filters to according to selection from the SRU, to produce a grounding heatmap on the original input image.

Since training a dual-path model implicitly trains the visual path in object localisation, and the presence of text metadata in archival datasets can be leveraged as a set of visual noun concepts [44], we propose to rethink the Visual Grounding task as a multi-class classification problem where training a dual-path model acts as a proxy task for object localisation. Furthermore, for Image Retrieval, text-guided retrieval using a trained localisation module can act as a text-based guide for improved

convolutional pooling in the Image Retrieval pipeline.

Chapter 4

CNN-Based Image Retrieval

Introduction

Content Based Image Retrieval is an image-to-image retrieval process, whereby an image is submitted as a query and images from a gallery/database are returned as the result. The intent of the Image Retrieval system is that the database images are sorted by some similarity to the query by the image's visual content, from 'most similar' to 'least similar'.

It would be desirable for such an Image Retrieval system to work in a high level, to sort according to the presence of objects or scenery. In the past, algorithms were handcrafted to capture low-level information such as colour and shapes, but not such high-level semantics. Hand-crafted visual features [11, 23, 67] can match unique local structures, known as keypoints, even at different scales and orientations, and are robust to minor affine transformations. However, these algorithms suffer from general drawbacks. They are only useful for finding common objects viewed from the same orientation, and are thus not as useful for finding similar objects or scenes in different perspective. Furthermore, image representations are high-dimensional and the algorithms are computationally heavy to implement.

With the advent of GPU-acceleration of Convolutional Neural Networks (CNNs) (Figure 2.8) and publicly-available pre-trained (or 'off-the-shelf') models, CNNs have

become a state-of-the-art tool for the Image Retrieval task. CNNs can extract from images high-level information and output fairly compact representations. CNNs trained on the large ImageNet dataset [24] are semantically-aware due to their training on object classes, and their multi-layer structure allows for features to be extracted at any layer that is desired. Some CNNs (*e.g.* VGG_{16} [103]) consist of two main modules: a Fully-Convolutional part and a Fully-Connected part. Output features from the latter Fully-Connected part can represent the global input image a single compact descriptor, and the output of the Fully-Convolutional part even maintains local spatial information that can be further pooled into a global image feature. Image Retrieval rankings work via highly discriminative but compact global image features (vectors) that allow for rapid similarity sorting using euclidean distance measurements via efficient matrix operations.

Post-processing descriptors can further boost performance, including feature whitening and the diffusion process [27]. Whitening is a step to reduce correlation between features. The diffusion step then models more complex structures in the vector space; while the euclidean distance measurement between image vectors works well, the actual descriptors can produce complex manifold structures in the high-dimensional descriptor space that are ignored. The diffusion process could further boost performance by employing the connections within these manifolds.

The remainder of this chapter is laid out as follows: the datasets used in Image Retrieval experiments are explained, then the concepts of mean average precision, whitening, and the diffusion process are described. the CNN for Image Retrieval is outlined to explain how the network can be used to produce vector representations of images, then a series of experiments are carried out on baseline datasets to test the different methodologies. Firstly the 'neural codes' [10] of the Fully-Connected layers are explored, then we delve into the convolutional pooling techniques, including SPoC [9] and CroW [48], and propose a simple improvement to the SPoC method to enhance retrieval, and analyse the effectiveness of the properties of CroW. We then describe the steps of diffusion and query expansion. We perform a comprehensive analysis of the retrieval performance of the NAA29k archival image dataset using the different pooling techniques and examine Image Retrieval performance on the

archival dataset.

Methodology

Any tool that makes predictions (outputs) on input data can be referred to as a *model*. A CNN model can be generally considered, for the purposes of these experiments, a black box (or series of boxes), or a complex function with many millions of parameters. Figure 2.8 shows an example CNN trained for image classification, containing a convolution module and a fully-connected module. While the final output (the softmax layer) is designed to output class-wise information, it can still be used to represent a global image feature. Furthermore, outputs from any preceding layer can be used as an image-level output. We intend here to use off-the-shelf ImageNet-trained [24] classification networks to perform Image Retrieval on a set of datasets, including the archival NAA29k set.

Image Retrieval Datasets

To measure the performance of the experiments in this chapter, we will use a number of benchmark Image Retrieval datasets:

- **Oxford5k** The Oxford5k dataset [89] consists of a gallery of 5,063 images of buildings and locations around the University of Oxford in the United Kingdom. It contains 55 cropped queries in groups of 11 (five images per building) and a groundtruth set of 'correct' images in the gallery.

- **Paris6k** Similar to Oxford5k, Paris6k [90] contains 6,093 images and also 55 queries, from buildings and locations in the city of Paris, France.

- **Holidays** The INRIA Holidays [46] dataset contains 1,491 scene-heavy holiday photographs. There are 500 small groups, with one query per group. Following [9] any incorrectly-rotated images have been manually rotated to the correct orientation.

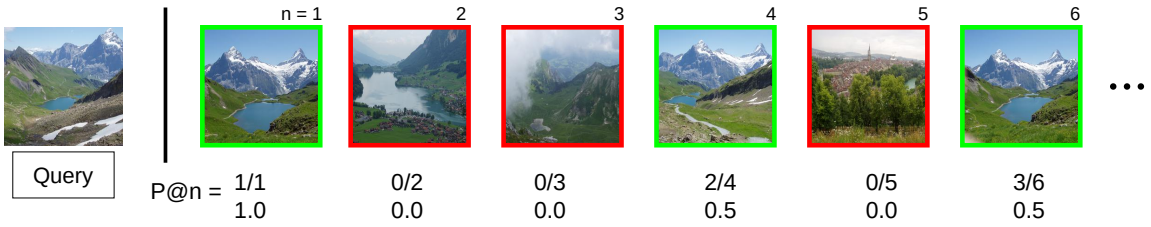

Figure 4.1: Example of a query and ranked results with three positive groundtruth images. Positive images are bordered in green, and negative images in red.

- **NAA29k** The NAA29k dataset is a collection of 28,912 images from the National Archives of Australia [5], as introduced in National Archives of Australia - NAA29k Dataset. Two sets of groundtruth were created: NAA29k_{100} and NAA29k_{1137}, which respectively have 100 and 1137 queries.

Mean Average Precision

Image Retrieval performance is evaluated using the mean-average-precision (mAP) metric. mAP is useful as a score for retrieval performance because each query usually has a small number of positive groundtruth images within a relatively large gallery. mAP rewards the model when its gallery has groundtruth images ranked highly (and penalises when they are ranked low) even if there are only a handful of them. Consider a dataset with a groundtruth consisting a set of queries $Q = \{q_1, q_2, ..., q_m\}$. For illustration, imagine the first query (q_1) has three positive groundtruth images in the gallery, and a model returns the ranks as shown in Figure 4.1, where the positive images occur in positions 1, 4, and 6.

The precision at each index (P@n) is calculated:

$$\text{P@}n = \frac{\text{relev}(i(n))}{n} \tag{4.1}$$

where $i(n)$ is the ranked image at position n and relev() is a function that is the number of true positive results seen so far *if* $i(n)$ *is true positive*, or zero otherwise.

The Average Precision for this query is simply the average of these precision

values:

$$\text{AP} = \frac{1}{|n|} \sum_{n=1}^{N} \text{P@}n \tag{4.2}$$

where N is the number of gallery images. The Average Precision is calculated for *each query* in the dataset's groundtruth, and finally, the Mean Average Precision is simply the mean of the Average Precision values:

$$\text{mAP} = \frac{1}{|Q|} \sum_{q=1}^{m} \text{AP}_q \tag{4.3}$$

The mAP provides a measure for the performance of an image retrieval model, and will be used in this chapter for all image retrieval datasets.

Models

All experiments are conducted in the Python machine learning library PyTorch [85], version 1.0.1. The pretrained VGG_{16} and VGG_{19} [103] models and the three largest ResNet [38] models are compared: ResNet_{50}, ResNet_{101}, and ResNet_{152}.

The VGG_{16} and VGG_{19} models consist of two main sub-networks: the Fully-Convolutional Network (FCN) and a Fully-Connected (FC) classification network. The FCN is made up of alternating convolutional layers and pooling layers. We are interested in the highly-semantic information in the latter layers of the networks. The final 1000-D output will be referred to as Softmax, the penultimate FC layer as FC_2, and the previous layer as FC_1. In the FCN, the final layer is a pooling layer called Pool_5, the penultimate layer is a convolution layer Conv_{5-4}, and the second-last layer is a convolution layer Conv_{5-3}.

The ResNet_{50}, ResNet_{101}, and ResNet_{152} models consist of 50, 101, and 152 convolutional layers, respectively. The major difference of the ResNet models is the introduction of skip connections between early and later layers.

Before being fed into the CNN, all images are resized to 256×256 pixels, and the average pixel value subtracted for stability. The image is a 3-D tensor $\boldsymbol{I} \in \mathbb{R}^{3\times256\times256}$.

Convolutional Pooling Methods

The output of a convolutional (or pooling) layer of a CNN produces a 3-D tensor $\boldsymbol{X} \in \mathbb{R}^{C \times H \times W}$. It contains $H \times W$ deep features of length C. Alternatively, it contains C channels of $H \times W$. The c_{th} channel can be represented as X_c. The output of each method is l_2-normalised.

MaxPooling

Max pooling performs a max() function over the C channels to produce an output that contains at each position the maximum value in that channel from all spatial locations to produce $Y \in \mathbb{R}^C$.

SumPooling

Sum pooling performs a summation of each spatial location on each channel

$$Y_i = \sum_{w=1}^{W} \sum_{h=1}^{H} X_i \tag{4.4}$$

and the final vector $Y \in \mathbb{R}^C$ is the concatenation of the summed channels:

$$Y = [Y_1, Y_2, ..., Y_C] \tag{4.5}$$

SPoC: SumPooled Convolutional Features[9]

The SPoC method is a variation of the SumPool method, whereby a parameter σ is chosen to affect a gaussian weighting across the spatial locations of the output tensor $\boldsymbol{X}$. The output of each spatial location $\boldsymbol{X}_{i(x,y)}$ is weighted by

$$w_{(x,y)} = exp\left(-\frac{(y - \frac{H}{2})^2 + (x - \frac{W}{2})^2}{2\sigma^2}\right) \tag{4.6}$$

with σ chosen as a constant of one-third the width W (or height H, which are the same in this case). Each spatial position (x, y) is weighted with multiplication by $w_{(x,y)}$ and SumPooling is performed.

CroW: Cross-Dimensional Weighting [48]

The SPoC algorithm leverages only the spatial information to weigh the deep features in order to improve performance. Cross-Dimensional Weighting (CroW) extends this idea further by introducing a channel weighting. The spatial weighting begins with a simple summation step over all channels of the tensor to produce a heatmap $\boldsymbol{S'}$:

$$\boldsymbol{S'} = \sum_{i=1}^{C} X_i \tag{4.7}$$

where S_i is the i^{th} feature channel in the tensor, and C is the number of channels in the tensor. A normalised heatmap $\hat{\boldsymbol{S}}$ is produced:

$$\hat{\boldsymbol{S}} = \left(\sum_{x,y} \sqrt{\boldsymbol{S'}_{x,y}} \right)^2 \tag{4.8}$$

Each spatial location (x, y) of the heatmap is then modified by dividing each spatial location of the heatmap by $\hat{\boldsymbol{S}}$ and square rooting:

$$S_{x,y} = \sqrt{\left(\frac{\boldsymbol{S'}_{x,y}}{\hat{\boldsymbol{S}}} \right)} \tag{4.9}$$

Each deep feature at position (x, y) can therefore be weighted with the weight $S_{x,y}$.

The channel weighting $Q \in \mathbb{R}^C$ is produced by analysing the output tensor and determining a sparsity value for each channel. Given that rarer features may correspond to visual features on important objects, it is possible to boost the values of rare features and suppress common features. Each element of the sparsity vector Q_i is calculated as a percentage of activations in the corresponding channel in excess of

a threshold (in this case, zero). The element at channel c is calculated:

$$Q_c = \frac{1}{WH} \sum_{x,y} \mathbb{1}[\boldsymbol{X}_{x,y,c} > 0] \tag{4.10}$$

where $\mathbb{1}$ is an indicator function counting a 1 where threshold is breached. The final boosting vector $I \in \mathbb{R}^C$ is produced:

$$I_c = \log\left(\frac{C\epsilon + \sum_h Q_h}{\epsilon + Q_c}\right) \tag{4.11}$$

where ϵ is a small positive value.

R-MAC [110]

Regional Maximum Activation of Convolutions (R-MAC) [110] divides the tensor $\boldsymbol{X}$ into a series of spatial regions depending on a layer parameter L. With $L = 0$ there is a single region that consists the entire tensor. At each subsequent layer $l_i, i > 0$ the tensor is cropped with equal regions of width $\frac{2\min(W,H)}{l+1}$. There is a defined overlap of 0.4 widths for each crop on the same level. We use $L = 3$ for these experiments following [110].

Feature Whitening

Dimensionality reduction on image vectors has been used to produce more concise descriptors to reduce storage, decrease computational overhead, or even improve performance [9, 18, 47, 48, 110]. This is performed using Principal Component Analysis (PCA). A related step to PCA is *whitening*, which may (or may not) involve dimensionality reduction, but ensures the features are uncorrelated, and have the same variance. To whiten an $n \times m$ matrix F (having n images and feature length m), begin by l_2-normalising F:

$$F := \frac{F}{||F||_2} \tag{4.12}$$

then calculate the covariance matrix Σ efficiently to produce a symmetric positive semi-definite matrix:

$$\Sigma = \frac{FF^{\mathrm{T}}}{m} \tag{4.13}$$

then perform Singular Value Decomposition (SVD) on the Σ to get the eigenvectors U and eigenvalues S. The PCA-whitened matrix F_w is computed by

$$F_w = \frac{1}{\sqrt{S+e}} U^T F \tag{4.14}$$

where e is a small number for numerical stability. Finally, the whitened matrix is l_2-normalised:

$$F_w := \frac{F_w}{||F_w||_2} \tag{4.15}$$

This whitening process follows the direction of [9, 48, 110] where features are firstly l_2-normalised, whitened, then l_2-normalised again.

Diffusion Process

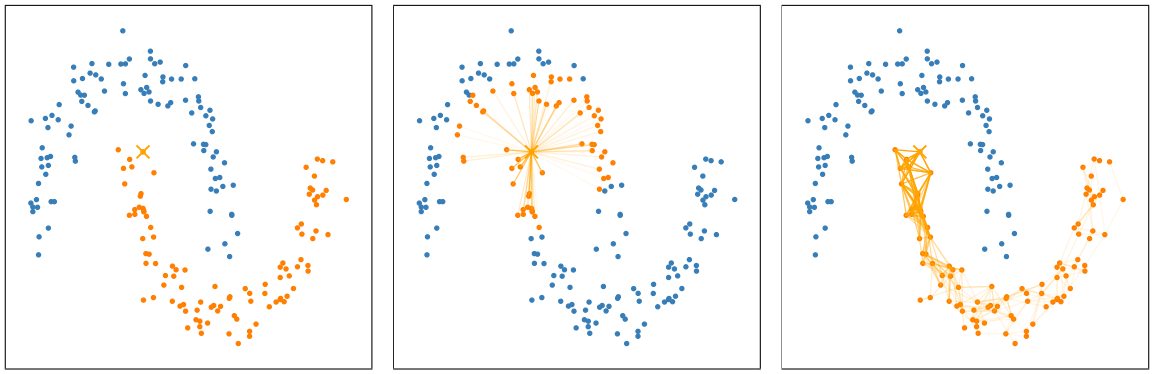

Figure 4.2: Illustration of the advantage of the diffusion process. In a two-dimensional toy example, two intertwining manifold structures of datapoints exist with a query denoted with an X (left). The datapoints in the orange manifold are distinct from those in the blue manifold, so the top-ranked results of querying X should contain only those in the orange manifold. Simply taking euclidean distance (center) includes datapoints from the wrong manifold, producing poor retrieval results. The diffusion process (right) diffuses similarity across the manifold to produce superior results. Best viewed in colour.

When performing Image Retrieval ranking it is computationally efficient to use an *affinity matrix*, *i.e.* a similarity matrix, that models the similarity of every pair of image descriptors. For a query the similarities are sorted from largest to smallest to produce query rankings. However, the similarity calculations performed in the high-dimensional space occur in distinct pairs, and do not consider further structure within the space. Such structures can include manifolds, whereby apparently distant pairs of image descriptors are actually related through latent connections via other intermediate descriptors across the manifold (Figure 4.2).

Intuitively, re-think the affinity matrix as an undirected graph, with the descriptors as nodes and their pairwise similarities as edges. The purpose of a diffusion process is to disperse the similarity information (edge weights) to neighbouring edges. The diffusion process is sometimes referred to as a random walk along the graph [27].

The diffusion process can be as follows: Consider an $N \times N$ affinity matrix $\mathbf{A}$, where N is the number of images in the dataset. $\mathbf{A}$ represents the similarities between each pair. $\mathbf{D}$ is an $N \times N$ diagonal matrix with the diagonal elements containing the row-wise sums of $\mathbf{A}$:

$$d_{i,j} = \begin{cases} deg(i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

where $deg(i)$ is the degree function (the sum of the edge weights of element i).

A *random walk transition matrix* $\mathbf{P}$ is then produced:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{A} \tag{4.17}$$

Since this intuitively represents a random walk of the graph, we can create a probability vector f_t of length N that represents the probability of being on each node at time step t. With this, an iterative update of the probability vector can be produced. At time zero ($t = 0$), f_0 represents some initial distribution, and the random walk is modeled as:

$$f_{t+1} = f_t\mathbf{P} \tag{4.18}$$

An additional 'random' movement across the graph can be included to model the

probability that there is a random link between two unrelated nodes. If there is a high probability α that the random walk follows the iterative update (Equation 4.18), then there is a $1 - \alpha$ chance the random jump can occur. Therefore, the iterative update rule can be expanded as

$$f_{t+1} = \alpha f_t \mathbf{P} + (1 - \alpha) y \tag{4.19}$$

where y is a 1-vector of length N.

Experiment: Fully Connected Codes

Convolutional Neural Networks for image classification can be generally divided into two main parts: the convolution block and the fully-connected (classification) block. The convolution block consists of convolution layers that successively extract higher levels of information from the previous block [78, 110, 128]. Although the latter fully-connected block of a classification network is trained specifically for class selection, the outputs of the intermediate hidden layers of the fully-connected block can be used as global image features [10].

This experiment will demonstrate the effectiveness of each of the fully-connected layer outputs as global image feature vectors for image retrieval. For comprehensiveness, five pre-trained models are used: VGG_{16} and VGG_{19} [103], and three ResNet models [38] ResNet_{50}, ResNet_{101}, and ResNet_{152}. The residual network differ in their internal construction, consisting of convolutional with long-term connections. They also feature only a single fully-connected layer at the end for the purposes of ImageNet [24] classification.

Model	Layer	Oxford5k	Paris6k	Holidays	NAA_{100}	NAA_{1137}
VGG_{16}	FC_1	**47.33**	61.90	**78.0**	**38.34**	**34.05**
	FC_2	42.89	57.73	76.92	33.64	30.27
	Softmax	35.34	55.84	71.31	28.07	24.27
VGG_{19}	FC_1	46.83	**64.67**	77.66	37.17	32.28
	FC_2	42.29	63.23	76.44	33.65	28.52
	Softmax	36.08	56.39	70.61	27.64	22.92
ResNet50	Softmax	34.86	55.20	75.35	27.64	22.92
ResNet101	Softmax	36.30	60.05	77.18	33.38	29.23
ResNet152	Softmax	39.79	60.13	77.63	31.53	28.28

Table 4.1: Results (in mAP) for Image Retrieval datasets using outputs from fully-connected layers. FC_1 refers to the second-last FC layer and FC_2 refers to the last layer before Softmax. The best result for each dataset is in bold.

Fully Connected Results and Discussion

The final output of the model (1000-length softmax outputs) are consistently lower than the FC layers. This is because the softmax is not designed as a global image feature, but rather a set of confidence scores for each of the 1000 ImageNet [24] classes.

The FC_1 layer of VGG_{16} performs best across the datasets, and the VGG_{16} produced improved descriptors on every dataset except for Paris6k, where VGG_{19} FC_1 outperformed VGG_{16} FC_1 by 2.77%.

On $NAA29k_{100}$ the VGG_{16} outputs outperformed ResNet by a minimum of 4.96%.

Some NAA29k queries are shown in Figure 4.3 and their top-10 retrieved images. Note that images of the same scene are collected well, and even similar-looking scenes that are technically incorrect are highly retrieved.

Clearly the VGG_{16} model shows the best performance compared to the other models tested. We will continue to use the VGG_{16} model for the following experiments. To highlight the benefit of the whitening and diffusion steps, outputted descriptors from the VGG_{16} model are further enhanced using whitening and dif-

fusion steps. We use the FC_1 and FC_2 layers and ignore the Softmax layer, which produced the lowest mAP performance for all datasets. We perform whitening separately, diffusion separately, and both whitening followed by diffusion. The results are shown in Table 4.2.

Figure 4.3: Fifteen queries from NAA29k_{100} and their top 10 retrieved images on VGG_{16} from the FC_1 output. The first column contains each query, and the following images are the top 10 results. Correct results are bordered in green, and incorrect results are bordered in red. Best viewed in colour.

Dataset	Layer	Baseline	Whitening	Diffusion	White+Diff
Oxford5k	FC_1	47.33	45.78	51.21	**61.32**
	FC_2	42.89	45.76	46.37	59.65
Paris6k	FC_1	61.90	41.81	70.87	**76.16**
	FC_2	57.73	45.35	67.81	74.03
Holidays	FC_1	78.00	79.49	79.82	**83.68**
	FC_2	76.92	77.41	77.72	82.38
$NAA29k_{100}$	FC_1	38.34	44.90	39.41	**46.60**
	FC_2	33.64	43.76	34.08	45.71
$NAA29k_{1137}$	FC_1	34.05	39.75	35.28	**41.81**
	FC_2	30.27	38.44	31.08	40.74

Table 4.2: Mean Average Precision results of Fully Connected descriptors from the VGG_{16} model. All descriptors are dimension-reduced to 512 dimensions. Results include after whitening only, after diffusion only, and after both whitening and diffusion process.

In all datasets the best result was achieved when performing both whitening and diffusion post-processing steps after the feature extraction. Compared to the baseline Oxford5k achieved an additional 13.99% and 14.26% on the FC_1 baseline. While there was less improvement on the Holidays dataset, that already had a higher mAP result. Intuitively, this is due to Holidays containing a large number of unconnected groups of images, and many groups of images having little visual relationship.

NAA29k benefited from both whitening and diffusion but gained more benefit from the whitening step. The $NAA29k_{100}$ groundtruth demonstrated a 8.26% improvement over the baseline feature extraction from both whitening and diffusion, and $NAA29k_{1137}$ was improved by 7.76%.

Experiment: Convolutional Pooling

The outputs from the FC layers in the classification module of a CNN can produce useful global image descriptors. Despite this simplicity, there are some limitations to using the FC or softmax outputs as descriptors. The input image must be of a specific size to ensure the input to the classification module is the correct shape, and

because the fully-connected weights require the inputs of the final pooling layer from the FCN, spatial information is not retained. This means that image regions cannot be extracted, as in the case of the R-MAC method.

Convolution pooling has advantages of producing more compact descriptors [9, 47] compared to the outputs of the fully-connected layers [10], and require less computation due to the removal of the classification part of the network. Furthermore, due to the stacked nature of the convolution and pooling layers in the FCN part of the network, the output tensor contains spatially-aware information from the input image. In this experiment we carry out the various convolution pooling methods described earlier. We will run each pooling method followed by l_2-normalisation but without whitening or other post-processing steps (as in [9, 48, 110]), to better compare specifically the pooling methods only.

We will use the FCN part of the pre-trained VGG_{16} network as the feature extractor and take the 3-D tensor $\boldsymbol{X}$. Each input image is first spatially resized to a tensor $\boldsymbol{I} \in \mathbb{R}^{3\times256\times256}$, and then the average pixel value subtracted. The output is taken from the final pooling layer Pool_5, and the previous convolution layers Conv_{5-4} and Conv_{5-3}.

Method	Layer	Oxford5k	Paris6k	Holidays	NAA_{100}	NAA_{1137}
MaxPool	$Pool_5$	45.72	63.75	77.07	36.06	30.63
	$Conv_{5-4}$	45.72	63.75	77.07	36.06	30.63
	$Conv_{5-3}$	44.06	62.42	76.35	35.18	30.15
SumPool	$Pool_5$	46.33	61.39	**80.80**	**41.50**	**35.03**
	$Conv_{5-4}$	44.04	59.15	79.75	40.04	33.83
	$Conv_{5-3}$	44.80	52.64	78.33	38.06	32.33
SPoC	$Pool_5$	46.22	60.18	78.98	40.03	33.78
	$Conv_{5-4}$	42.12	53.00	76.23	37.55	31.23
	$Conv_{5-3}$	40.56	51.18	75.63	35.47	30.87
CroW	$Pool_5$	45.88	**64.23**	75.70	34.47	29.41
	$Conv_{5-4}$	45.44	64.12	74.61	34.28	28.78
	$Conv_{5-3}$	**46.85**	55.78	76.09	35.14	30.36
R-MAC	$Pool_5$	45.12	60.15	80.53	38.88	33.96
	$Conv_{5-4}$	45.53	60.45	80.72	39.16	33.96
	$Conv_{5-3}$	44.61	63.13	79.02	36.90	31.74

Table 4.3: Mean Average Precision results for the five datasets using the convolution pooling methods on the VGG_{16} model. Best results are highlighted in bold.

The experimental results are shown in Table 4.3, and show that the simpler SumPooling method produces the best mAP results for the Holidays and NAA29k datasets. Note that for the MaxPool method the retrieval results for the $Pool_5$ and $Conv_{5-4}$ results are identical across all datasets because the $Pool_5$ layer is simply a pooled max output, and performing the max operation on both produced identical image descriptors. MaxPooling did not achieve top results on any dataset. We attribute this to an information loss, as only the top spatial activation of each image is retained and the rest discarded, making it especially less suitable for scene-heavy datasets.

CroW managed to outperform SumPooling for the Oxford5k and Paris6k datasets, while SumPooling achieved top results for Holidays and the archival set NAA29k. Oxford5k and Paris6k are object-heavy and appear to benefit somewhat from CroW's

channel and spatial weights.

We believe the strength of the SumPooling method on the scene-heavy datasets is its addition of all spatial features in an unweighted manner. This provides an advantage on Holidays, where useful features are spatially dispersed in visual scenes, and background features are just as important as foreground objects. This is why simpler SumPooling outperforms SPoC with the center weighting on Holidays - the center weight would reduce influence of the global scene and focus more on a smaller collection of visual features.

For the scene-heavy Holidays dataset the R-MAC method generally improved over the other methods except for SumPool. We suspect this is the case because the R-MAC method takes in more information of the scene at different scales. For the object-heavy datasets we suspect it underperformed due to including too much irrelevant background features instead of important center-positioned objects.

The CroW method produced top results for Oxford5k and Paris6k datasets, while SPoC did not produce best results on any of the datasets. In the next section we explore the individual spatial/channel components of the CroW method to assess their importance, and in the following section propose a simple manner to test the optimum SPoC σ parameter for each dataset.

CroW Analysis

The CroW method, is a SumPooling operation that includes two separate weighting schemes: a channel weighting and a spatial weighting. We see from the previous section that CroW produced best results for Oxford5k and Paris6k datasets, but not for the other scene-heavy datasets. Here we desire to understand more about how the weighting affects the retrieval results across the datasets.

In this experiment we run the CroW method but in the 'Spatial Only' mode set the channel weights to be 1.0, and in the 'Channel Only' mode set the spatial weights to be 1.0. This will produce descriptors for each weighting scheme separately. Retrieval results in mAP are shown in Table 4.4.

Method	Layer	Oxford5k	Paris6k	Holidays	NAA_{100}	NAA_{1137}
No Weight	$Pool_5$	46.33	61.39	80.80	41.50	35.03
	$Conv_{5-4}$	44.04	59.15	79.75	40.04	33.83
	$Conv_{5-3}$	44.80	52.64	78.33	38.06	32.33
Full CroW	$Pool_5$	45.88	**64.23**	75.70	34.47	29.41
	$Conv_{5-4}$	45.44	64.12	74.61	34.28	28.78
	$Conv_{5-3}$	46.85	55.78	76.09	35.14	30.36
Spatial Only	$Pool_5$	46.18	**64.23**	**81.61**	35.89	30.27
	$Conv_{5-4}$	45.20	63.62	75.52	35.01	29.31
	$Conv_{5-3}$	45.25	54.12	75.54	33.99	29.30
Channel Only	$Pool_5$	**47.63**	63.35	75.70	**42.37**	**35.76**
	$Conv_{5-4}$	46.21	62.32	81.01	41.94	35.33
	$Conv_{5-3}$	44.80	52.64	78.33	38.06	32.33

Table 4.4: Mean Average Precision of the full CroW algorithm, and alternatively using spatial and/or channel weighting schemes only.

Interestingly, we observe that for the Oxford5k and NAA29k datasets channel-only information produced improved descriptors compared to the full weighting scheme. Paris6k is largely unaffected by the removal of the channel information, and produces the same retrieval performance when channel weighting is removed from the $Pool_5$ layer. Channel-only weighting achieved a 1.75% improvement on Paris6k when using channel information only, compared to the full CroW method, but NAA29k improved far better at 7.9% for $NAA29k_{100}$ and 6.35% for $NAA29k_{1137}$.

Channel weighting is a self-weighting, calculated by the sparsity (rarity) of non-zero activations in a channel, so more activations producing a boosted weight at that channel. This increases existing high activations and suppresses low activations.

Holidays achieved 5.91% improvement on spatial-only over channel-only weighting. Holidays strongly benefits from the spatial weighting because that weighting scheme produces a full spatial map and normalises it, meaning it includes all provided spatial information. The Holidays is the most scene-heavy, and it is most useful

to utilise global image features. Removing channel weighting stops the suppression of useful channel-wise information.

We attribute the improvement on NAA29k dataset using channel-only weighting by its ability to boost strongly-detected features in each channel and suppress weakly-detected features. The weakness of the spatial weighting here is because it sums and normalises all deep features to produce a global importance heatmap. This is less effective for the archival set NAA29k, which contains significant image spatial occlusion and distractions Figure 2.4. Therefore for archival sets the channel weighting provides the best benefit by boosting useful channel-wise features.

Improving SPoC with Dataset Weighting

Sum-Pooled Convolution Features (SPoC) [9] is a SumPooling method that firstly produces a constant Gaussian weighting over the output tensor $\boldsymbol{X}$ before carrying out the SumPooling operation. This method intents to increase the activations of the deep features located toward the spatial center of the tensor, which should correspond to objects in the center of the input image. This method relies on the assumption that more interesting object features are towards the center of the image [9] in order to highlight those features more than irrelevant ones towards the edges of the image. Ultimately the σ parameter is determined by the tensor shape.

This hypothesis may be correct for object-heavy image datasets where the primary object of interest hlis in the middle. For Paris6k and Oxford5k datasets this is the case, but for scene-heavy datasets the positioning of images across the dataset may be more dispersed. Therefore, each dataset may have a unique spatial bias that produces a specific weighting scheme.

We propose a simple change to the weighting scheme that produces a new weighting heatmap for each dataset, which requires no human annotation, and does not require a σ parameter. We wish to use this to see the optimum weighting and how it can further improve the SPoC method. The only performance disadvantage is that it requires feeding the images in the feature extractor twice. For a tensor $\boldsymbol{X}$ of shape $C \times H \times W$ the Gaussian function (Equation 4.6) produces a weight matrix of size

$H \times W$ that is multiplied over the tensor on each channel.

Choosing a Best Sigma. Firstly we highlight that the arbitrary choice of σ in the Gaussian weighting operation can be sub-optimal. We exhaustively run the SPoC pooling operation on Oxford5k, Paris6k, Holidays, and NAA29k_{100} with σ value between 0.1 and 10.0 at 0.1 increments using the Pool_5 outputs of the VGG_{16} model, and graph their Mean Average Precision results in Figure 4.4. The Gaussian weighting function produces more acute center weighting when σ is low - as σ increases, the weight $\alpha_{(x,y)}$ tends to 1.0 for all (x, y).

It is clear that with low σ values ($\sigma < 1.0$) the weighting scheme places too much weighting to the exact center, and retrieval results are poor across all datasets. However, there is a clear peak in retrieval performance for the Oxford5k and Paris6k datasets where a specific σ value would benefit. The Holidays and NAA29k datasets did not show a specific peak, but higher sigma values plateaued the retrieval performance. We suspect that due to the scene-heavy nature of these two datasets they respond less to the Gaussian center-weighting scheme.

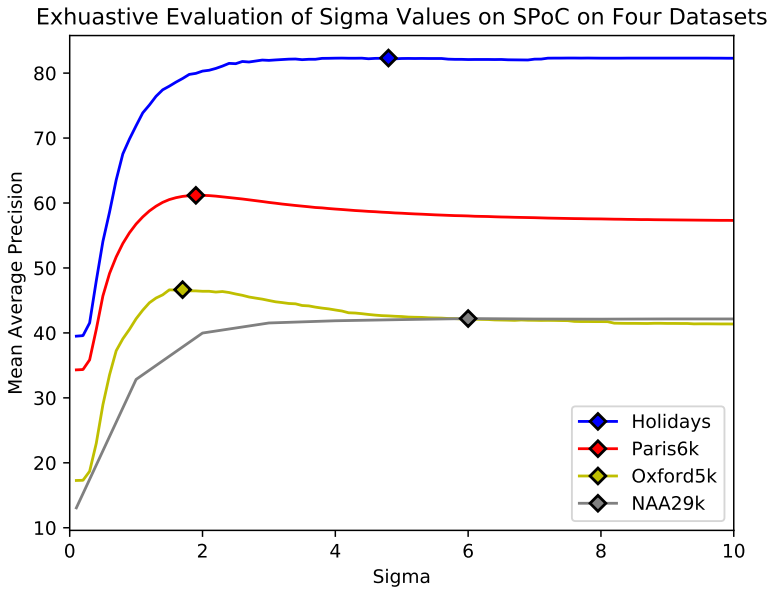

Figure 4.4: Repeating the SPoC centering algorithm with varying values for the hyperparameter sigma in the gaussian weighting function. The performance is measured in Mean Average Precision, and the highest performance for each of the four datasets is indicated with a diamond.

Producing the SPoC Weighting Heatmap. To produce a weighting heatmap for a dataset, each image $\boldsymbol{I}_i$ is fed through the feature extractor to produce tensor $\boldsymbol{X}_i \in \mathbb{R}^{C \times H \times W}$. The i^{th} image's output tensor is SumPooled along the channel dimension C to produce a 2-D heatmap Z_i:

$$Z_i = \sum_{c=0}^{C} X_{i,c} \tag{4.20}$$

where $X_{i,c}$ is the c^{th} channel of the i^{th} image, and $Z_i \in \mathbb{R}^{H \times W}$. The final heatmap is a summation of each image's heatmap:

$$Z = \sum_{i=0}^{N} Z_i \tag{4.21}$$

where there are N images in the dataset. The final heatmap is then l_2-normalised.

We use the VGG_{16} pre-trained network as before as the feature extractor, input images of size 256×256, and produce the weighting heatmaps (of size 8×8) for the Oxford5k, Paris6k, Holidays, and NAA29k datasets. They are displayed in Figure 4.5. Redder activations correspond to high activation values, while blue represent lower activations. Oxford5k clearly has a center-focused dataset, with low values along the edges and high activation at the center. This corresponds with the peak in Figure 4.4. NAA29k and Holidays both show more semantic features in the corners, while Paris6k shows more information on the bottom edge.

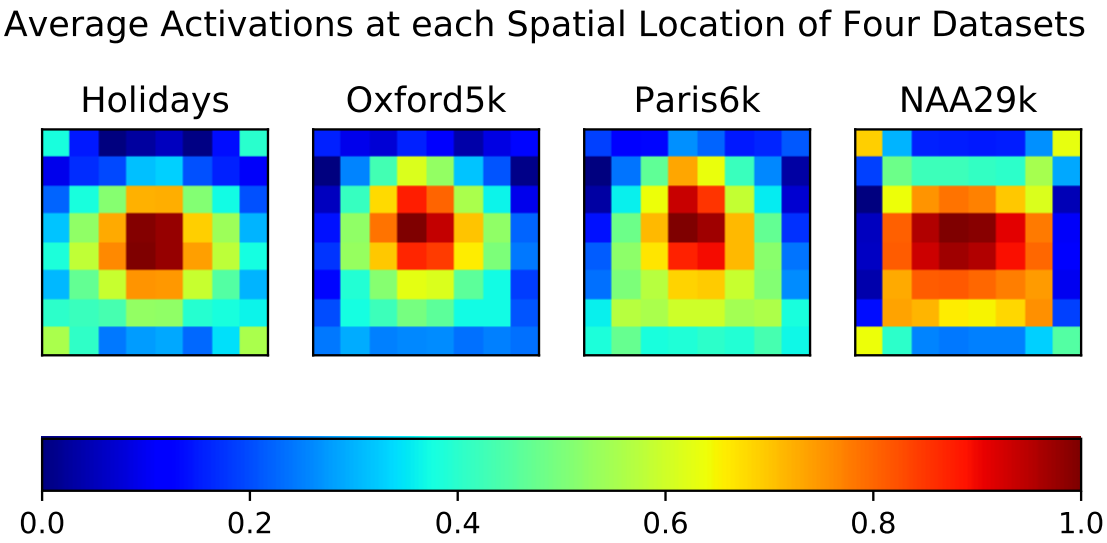

Figure 4.5: Heatmaps of the average activations overall all spatial dimensions of all images in the datasets Oxford5k, Paris6k, Holidays, and NAA29k. The heatmaps visualise that across all datasets the objects and semantic features are mostly focused on the center of the images.

The heatmaps are used in place of the original SPoC Gaussian weighting scheme and results compared in Table 4.5.

Method	Layer	Oxford5k	Paris6k	Holidays	NAA_{100}	NAA_{1137}
$SPoC_{sigma}$	$Pool_5$	46.22	60.18	78.98	40.03	33.78
	$Conv_{5-4}$	32.12	53.00	76.23	37.55	31.23
	$Conv_{5-3}$	40.56	51.18	75.63	35.47	30.87
$SPoC_{sigma\_best}$	$Pool_5$	**48.47**	62.68	80.29	**41.44**	35.19
	$Conv_{5-4}$	33.61	54.40	78.20	38.71	33.22
	$Conv_{5-3}$	42.11	52.58	77.63	37.94	33.01
$SPoC_{heatmap}$	$Pool_5$	46.92	**63.79**	**81.09**	41.26	**35.30**
	$Conv_{5-4}$	44.88	62.01	79.93	40.04	34.19
	$Conv_{5-3}$	45.66	53.94	78.44	38.16	32.48

Table 4.5: Comparison of the standard SPoC with constant σ, the SPoC with exhaustive search for best σ, and SPoC with proposed heatmap weighting.

The proposed heatmap scheme outperforms the best Gaussian values for the Paris6k and Holidays datasets, and the NAA29k dataset with the 1137-query groundtruth. The Oxford5k dataset performed best using the Gaussian weighting function, with an increase of 1.55% against the proposed weighting, but the proposed weighting still exceeded the retrieval results using the original σ value in the Gaussian weighting function. The proposed heatmap scheme beat the original SPoC on $NAA29k_{100}$ and $NAA29k_{1137}$ by 1.23% and 1.52% respectively.

Whitening and Diffusion. Finally we demonstrate the post-processing steps of whitening and diffusion process on the output descriptors from the $Pool_5$ layer using the proposed heatmap scheme. Since diffusion-only without whitening produced poorer results than whitening followed by diffusion in the fully-connected output descriptors, we perform whitening, and white+diffusion, and show the results in Table 4.6.

Dataset	Baseline	Whitening	White + Diffusion
Oxford5k	46.92	46.99	**47.98**
Paris6k	63.79	62.49	**69.84**
Holidays	81.09	81.05	**81.30**
NAA29k_{100}	41.26	**41.91**	41.84
NAA29k_{1137}	35.30	**35.95**	35.92

Table 4.6: Effect of PCA whitening and diffusion on the baseline results of the $\text{SPoC}_{heatmap}$ features.

The NAA29k dataset increased retrieval performance with only whitening, while the diffusion step actually had a negative affect on performance. Paris6k received the best performance increase from the diffusion step, with a 7.35% increase, followed by Oxford5k with a 0.99% increase and Holidays with 0.25%. This is in contrast to the fully connected values which gained a greater boost in retrieval performance for NAA29k - the 100-query groundtruth reported a 7.19% increase in performance after the diffusion step.

Chapter Conclusion

In this chapter we have performed retrieval using the deep features from pretrained convolutional neural networks for Content Based Image Retrieval. We outlined four datasets that we used for image retrieval: Oxford5k, Paris6k, and INRIA Holidays datasets, and the archival NAA29k dataset.

Post-processing steps of feature whitening and the diffusion process can improve the retrieval performance by eliminating correlation between features, and tapping into the underlying structure of the high-dimensional descriptor space. The outputs of Fully-Connected layers require higher computation and produce descriptors of higher dimensions than outputs of the convolutional layers in the convolutional neural networks, but received enhanced retrieval performance when using post-processing steps.

We overviewed a series of convolutional pooling strategies that exploit the more

compact convolution output tensors. These were the MaxPool, SumPool, SPoC [9], CroW [48], and R-MAC [110] pooling schemes. The scene-heavy Holidays and NAA29k datasets benefited from simpler SumPooling operations while the other object-heavy datasets did not. The CroW pooling method produced good results on Oxford5k and Paris6k as compared to SumPooling, and a breakdown of its weighting schemes showed that retrieval on Paris6k was unaffected by its channel weighting. Our results conclude that the CroW's channel weighting is more effective for archival datasets by suppressing distracting channel features.

We proposed a simple improvement that replaces the 2-D Gaussian weighting function with a 2-D weight that requires no extra human input, or other hyperparameters. The proposed heatmap scheme outperformed the original SPoC on each dataset.

Content Based Image Retrieval can be adequately performed using pretrained convolutional neural networks as visual feature extractors, but as is the case for archival datasets like NAA29k, there is text metadata that can be exploited. In the next chapter we will explore the Visual Grounding task for the task of object localisation, as a stepping stone for text-guided retrieval.

Chapter 5

Visual Grounding Utilising Word2Vec Semantic Structure

Introduction

Visual Grounding [16, 26], also known as localisation, is a computer vision task that aims to train a model that can spatially localise a word or text phrase on an image. As we have seen in the previous chapter, pretrained Convolutional Neural Networks (CNNs) trained on classification can act as complex functions that project image data into a high-dimensional feature space suitable for the image retrieval task. Similarly, text information can also be embedded into a shared embedding space using recurrent neural networks [33, 49]. Such localisation models are usually trained with word vector inputs, sometimes from a pre-trained Word2Vec model, and then the new text module trained as one part of the localisation model [29]. As focus shifts to training weakly-supervised models without the need for expensive bounding box annotations in the training data, approaches include using singular word embeddings as part of the training phase, whereby localisation happens automatically as part of the model learning [80]. We are motivated to formulate weakly-supervised learning as a multi-class classification problem, using noun-words in the provided text annotations as classes. Using a pretrained Word2Vec model, we wish to determine whether

pretrained Word2Vec embeddings on a large vocabulary is sufficient for model training, without the need for a learnable recurrent module.

We develop a model for training and an offline stage that uses the Word2Vec word embeddings, while utilising the model's semantic structure for untrained words after training is completed. Our model includes a convolution layer that acts as an object detector, and we investigate the effectiveness of using multiple modules in parallel. Finally, we examine the performance using two modules in series with an erasure technique [132] to allow the detectors to discover different object features.

Related Work

Early form of class localisation included the Class Activation Mapping [96] that performs localisation of a finite number of visual classes by learning linear combinations of pooled feature maps from a FCN backbone. This weakly-supervised object localisation (WSOL) method relies only on image-level labels. Their method implicitly uses a linear combination of filters as object detectors. A problem with this approach is that the learned filter combinations only focus on the most discriminatory parts of each object class, such as the head of an animal. To capture more of the object's entirety [123] simply fuse a series of CAMs by considering combinations of high- and low-discriminative activations.

The current motivation is to perform weakly-supervised training without the need for bounding boxes [29, 65, 135] to eliminate the need for expensive annotations. Recent approaches involve multi-path deep learning models that embed multi-modal data into a shared embedding/semantic space. [49] embed Word2Vec [69] word embeddings using a bi-directional recurrent network [101] and a VGG network for the image path. Similarly, [29] use a Single Recurrent Unit (SRU) [59] to embed Word2Vec embeddings for the multi-modal retrieval task. Training a convolution layer as an object detector with labels implicitly produces an object localiser [80]. This technique is used to detect objects with sentences by isolating specific feature maps in the trained convolution layer to localise the object described in that text

[29]. With both images and text embedded into the same space, model learning can take place by calculating error between the embedded features and enforce euclidean distances. The triplet loss is used to enforce image embeddings and text embeddings to have small distance, while unrelated text embeddings have high distance in the semantic space [29, 115].

Another approach is to construct an encoder-decoder framework that jointly encodes both visual and text information into a joint space with a fully-connected neural network, then decodes those localisation heatmaps into text concept predictions [44]. They compel the network to learn by using batches containing the same visual concept. In all cases the joint embedding of image and text in a learned model acts as a means for localisation.

In each case a CNN acts as a powerful feature extractor while a following convolutional layer acts as a detector. Furthermore, boosting performance can occur with erasure/masking or using multi-detectors for part detection. Erasing input image information following the first feature detector can capture more of the object's entirety [117], or using two complementary convolutional feature detectors can detect different object regions [132].

We argue that since localisation is considered a task of localising short text phrases to related visual objects, visual concepts are largely interchangeable with noun-phrases [44]. While dual-path models can act as a proxy task for localisation [29], we argue that an encoder using only the structure of a Word2Vec model, without directly learning projected word embeddings, can still act as a powerful object localiser, even for concepts unseen at training time. We leverage an off-the-shelf part-of-speech detector with a pre-trained text vector model as a semantic gap-filler that not only allows our model to train on a relatively large concept vocabulary, but can still localise for unseen concepts.

The remainder of this chapter is as follows. We introduce the Word2Vec [69] text model and the two localisation datasets Flickr30k and MSCOCO, the loss function for model learning, and the Pointing Game metric. We then propose our encoder-based model architecture, and justify how we choose the number of trained classes. Finally, we show our experimental results of our model, and show improved perfor-

mance using multiple convolutional layers and the erasure technique [117].

Word2Vec Model

A Word2Vec model is a tool that directly transforms a word into a vector representation [68] [69], with the aim that words with close semantic meaning are close in the vector space. These semantic models map complex linguistic relationships between words that exhibit behaviours such as word clustering and relationships [69]. For our experiments we use the Google News Word2Vec [69] that embeds words into a 300-D vector space.

We are motivated to use a Word2Vec model to overcome some limitations of the simpler 1-hot method. In the 1-hot method, words are represented as a vector where all are zeros except for a 1.0 in one element. The limitation of this method is that the size of the word dictionary/vocabulary is fixed, and the length of the vector is equal to the vocabulary size, which can become very large. During model testing, there is the possibility of *unseen* word that is not in the vocabulary at training, and cannot be represented as a 1-hot vector. There is little difference in performance whether the text path is trained using 1-hot vectors or Word2Vec embeddings [49]. However, Word2Vec overcomes the vocabulary size problem because the Word2Vec embeddings are compact. The Word2Vec is structured so that words that have similar semantics have vectors that are closer in the embedding space by cosine similarity (see Figure 5.1). Therefore, in the case of an unseen word, a semantically-similar substitute can act as a placeholder during localisation.

We demonstrate this idea by selecting four words: 'tiger', 'sedan', 'bird', and 'skirt', and showing the top-10 closest words by cosine similarity in the Word2Vec [69] model. Observe that in the example of 'sedan', the next words are 'hatchback' and 'coupe'. In test situations where 'sedan' is not part of the training vocabulary, the nearest words are sufficiently semantically similar to act as placeholders.

There are several available pretrained Word2Vec models, but the initialisation of a Word2Vec model has little effect on overall performance, even when using a ran-

dom initialisation of word vectors [49]. In our preliminary experiments we determined whether the specific Word2Vec model has significant effect on the performance of our model learning. We tested the 300-D GloVe vectors [86], pre-trained Bert embeddings [25], the pre-trained Google Word2Vec model [69], and also on one-hot vectors for sanity. The Bert embeddings were extracted from the first layer of the Bert model. There was no material difference in performance in preliminary experiments, except for bad performance using one-hot, where each word is represented by a long vector whose length is the size of the vocabulary. The one-hot vector has no inherent semantic structure between words, and output on unseen words was seemingly random due to no underlying semantic organisation.

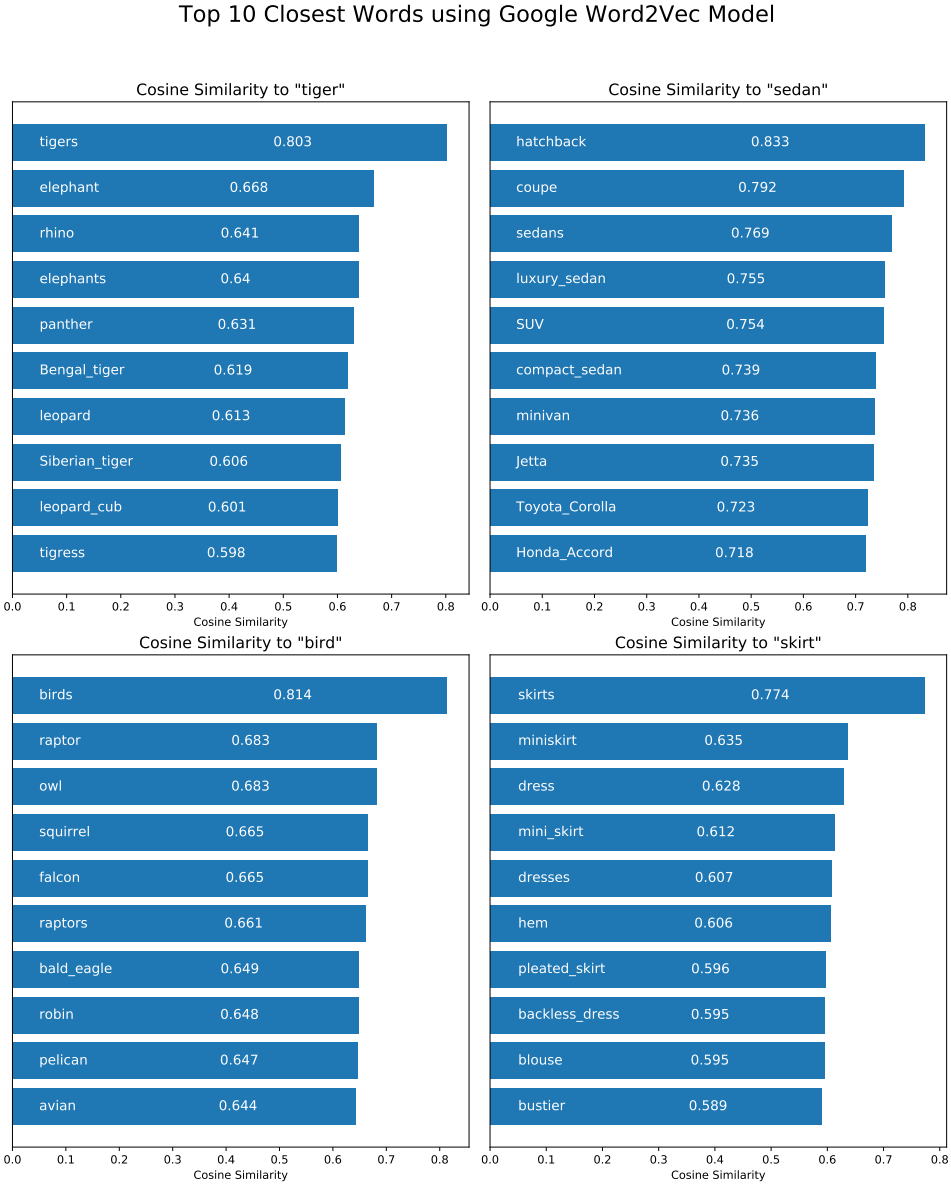

Figure 5.1: The top-10 words by cosine similarity to the words ‘tiger’, ‘bird’, and ‘sedan’ in the Google Word2Vec model.

Loss Functions

A loss function is a function that measures an error between the prediction of a model and the desired target considering the model's inputs. A model prediction that differs highly from the desired target produces a high error, while a model prediction close to the target delivers a low error value. The error value is backpropogated through the training model to update model weights to bring the output closer to the desired target. In this chapter we use the Cross-Entropy Loss function.

Cross-Entropy Loss

Cross-entropy loss aims to penalise the model when the predicted class differs from the target class. The loss value decreases when the predicted class is close to the target class. A model output in this case is a vector of length C where there are C possible classes and the i^{th} element of the vector represents the probability that the input is the i^{th} class. The cross-entropy loss of a model prediction p is

$$-\sum_{c=1}^{M} t_c \log(p_c) \tag{5.1}$$

where p_c is the probability that the input is class c, and t_c is a binary value that indicates if c is the correct class.

Datasets

Several datasets have been used in the visual grounding/localisation task, including Flickr30k [92], MSCOCO [62], Visual Genome [54], and ReferIt [50].

For our experiments we will utilise the MSCOCO and Flickr30k datasets in line with [7, 29, 31, 129] with the MSCOCO as training and Flickr30k for validation.

MSCOCO. The MSCOCO dataset utilised by [29] is made up of 123,287 images that are a combination of both the original training set (82,783 images) plus the validation set (40,504 images). This is because the Flickr30k dataset will be used

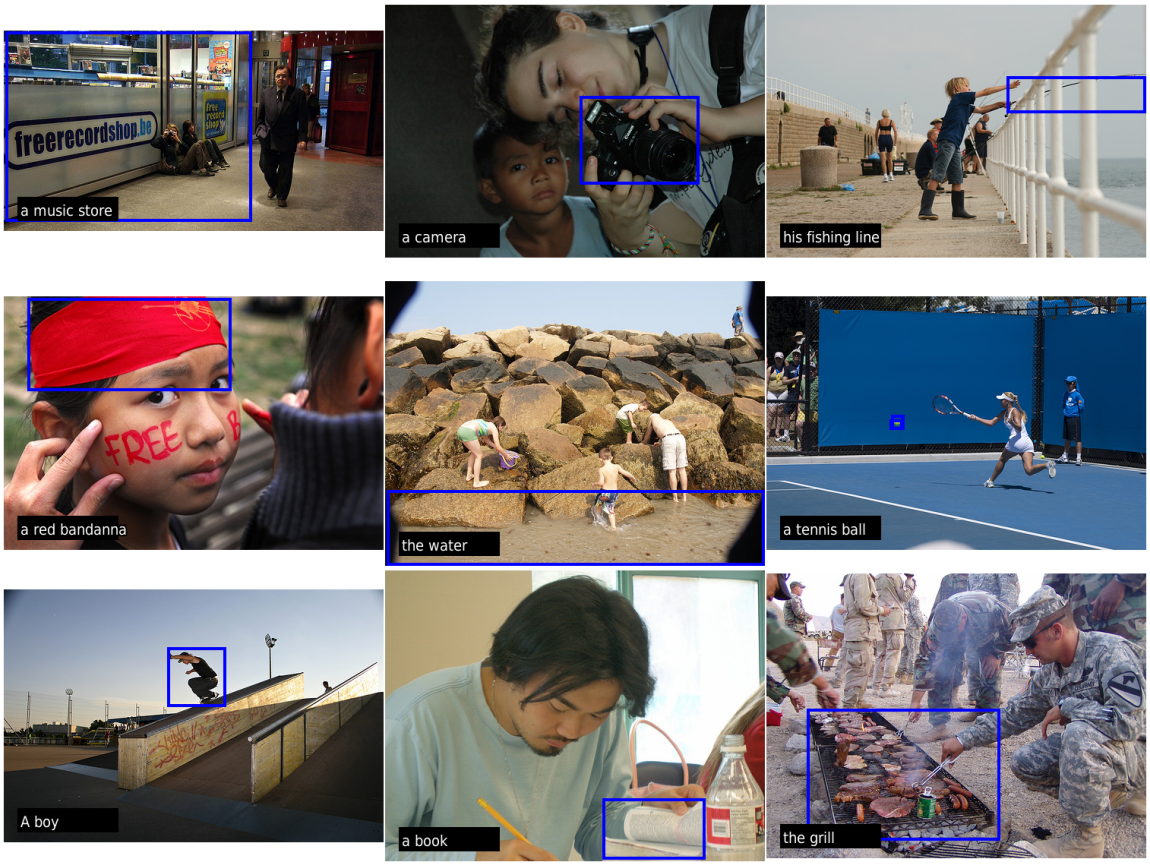

Figure 5.2: A selection of 9 images from Flick30k with one localisation phrase and its corresponding bounding box(es). Each bounding box is represented as a blue rectangle.

for validation. Each MSCOCO image has five associated captions, or phrases, that describe the image.

Flickr30k. The Flickr30k dataset contains 29,783 training images and 1000 validation images, as well as 1000 test images. All validation and test images have a set of bounding boxes that encompass visual objects in the image, and corresponding short phrases that are to be localised. The goal is to use the short phrases and image as input to a trained model, and localise the visual object that the phrase describes using the Pointing Game metric as described below. A small set of images and their accompanying phrases and bounding boxes are shown in Figure 5.2.

Pointing Game

The Pointing Game is a quantitative measurement of the effectiveness of a model on localising phrases to their respective bounding boxes in an input image. The trained localisation model should 'point' to a spatial position in an input image that best corresponds to the visual feature described by the input phrase. Each phrase is accompanied by groundtruth bounding boxes that envelopes the visual object(s). When the point occurs inside one of the bounding boxes coupled with the localisation phrase then it is considered a *#hit*, and if it is not then it is a *#miss*. The total pointing game score for one image is $\frac{\#hit}{\#hit+\#miss}$ and the total pointing game score is the percentage of all hits over all localisation phrases across every image.

Pointing Game Baselines

To establish baseline on the pointing game for the Flickr30k dataset, it is a simple task to 'simulate' the pointing using various techniques. Center means to always point at the image center, $(\frac{w}{2}, \frac{h}{2})$ where w is the original image width and h is the original image height, and this produces a pointing game accuracy of 49.20%. Random is a random (x, y) point on each image, and we show an accuracy of 27.24%. The random result differs each time the algorithm is run, so we ran it five times and took the average. VGG_{16} accuracy means to feed each image through a pre-trained VGG_{16} network only and take the spatial point that produces the highest activation in the output tensor. We ran it through the pre-trained VGG_{16} network with the classification and softmax layers removed, so the output was represented by the tensor output of the Pool_5 layer. This result produced a pointing game accuracy of 35.37%.

Proposed Architecture and Pipeline

Visual Pipeline

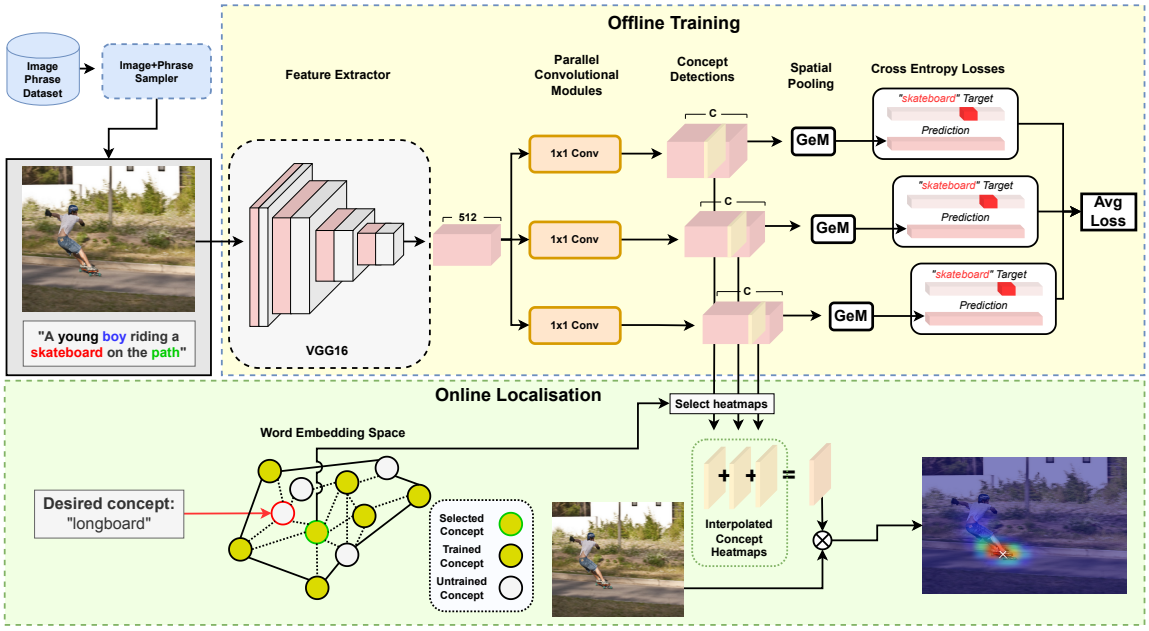

Figure 5.3: Overview of the proposed architecture and pipeline. The main image path contains a CNN such as VGG_{16} and a series of one or more convolutional layers in parallel that act as concept detectors. Separate spatial pooling modules produce predictions of object, and loss is calculated for one random concept known in the image phrase. In the online stage the user selects their desired concept or phrase, and using the pre-trained word space a best fit is found. Trained concept detectors output heatmaps corresponding to that concept.

We use a pre-trained CNN as our base network to act as a feature extractor. We choose the VGG_{16} network used in the previous chapter, and as used in [31, 44], and then remove the classifier and softmax to leave a FCN. The FCN accepts an input tensor $\boldsymbol{I} \in \mathbb{R}^{3\times256\times256}$ and the output of the FCN is a tensor $\boldsymbol{X} \in \mathbb{R}^{512\times H\times W}$.

One or more convolutional modules Φ_i take $\boldsymbol{X}$ as input, and output $\phi_i \in \mathbb{R}^{C\times H\times W}$ where C is the number of known concepts in the training dictionary/vocabulary. A learnable 1×1 convolutional kernel then produces confidence scores for each known

concept. A 1×1 convolutional kernel is a 3-D kernel of shape $C \times 1 \times 1$, meaning its receptive field has width and height of one. Its important characteristic is that the output tensor has equal shape to the input tensor, and it acts as a linear transformation of each spatial location of the output. This means that the kernel will output a confidence score reflecting the presence of a concept that kernel is intended to find.

Outputs from each convolutional module are spatially pooled using separate GeM [76] modules to produce C-dimensional predictions of object class, which are then passed through Softmax. Rectified Linear activation units (ReLU) are also used after VGG_{16} and each Φ_i but not shown for clarity.

Therefore, prediction p_i is $\text{Softmax}(\text{GeM}_i(\Phi_i(\text{VGG}_{16}(\boldsymbol{I}))))$.

Online Localisation

Intuitively, the visual model is trained to predict a target that is a one-hot vector with the j^{th} element a 1.0 (and all other elements 0.0) representing the j^{th} noun in the training vocabulary that is the object in the image. Therefore in the localisation stage the output of the j^{th} channel corresponds to high neuron activation upon seeing the visual object, and can be 'sliced' out for heatmap generation.

Sliced heatmaps are 2-D visual signals and can be interpolated to a common size if Φ output sizes vary, and are then combined with addition. To overlay heatmap, a final interpolation to size $3 \times 256 \times 256$ and multiplication with the original image $\boldsymbol{I}$ produces a heatmap localisation.

Loss Function

Cross-entropy loss is calculated on each prediction against a one-hot vector that represents the class target. For a C-dimensional one-hot target vector t and a prediction p_i from the i^{th} pooled parallel convolution output, the cross-entropy loss of is calculated as

$$L_i = -\sum_{c=1}^{C} t_c \log(p_c) \tag{5.2}$$

and the final loss is the average of all cross-entropy losses across $|L|$ parallel modules:

$$\mathcal{L} = \frac{\sum_{i=0}^{|L|} L}{N} \tag{5.3}$$

Justification for Object-Noun Threshold

For vocabulary selection, the training phrases of the train set are trawled and the noun words extracted using the NLTK [13] toolkit. For example, in the Flickr30k training set there are 11,707 nouns and in MSCOCO there are 13,068. Some are more common than others, and in fact there are 3,953 nouns in Flickr30k training set and 4,392 nouns in the MSCOCO training set that each occurs in only a single image. Therefore it is prudent to reduce the vocabulary size by removing nouns that exist in few images. By setting a threshold of how many images each noun exists in we can see how the vocabulary changes size. A set of thresholds and their respected vocabularies in Flickr30k and MSCOCO are shown in Table 5.1. Training a large vocabulary would increase training time and provide very few visual examples for some words. We intend to instead use the semantic structure of a Word2Vec model for such rare concepts.

Image Threshold	Number of Concepts	
	Flickr30k	MSCOCO
1	11707	13068
2	7754	8676
5	4866	5517
10	3492	3978
20	2416	2867
50	1352	1780
100	845	1200
200	475	777
500	205	450

Table 5.1: Image thresholds and the corresponding number of concepts that are found in the minimum number of images in the datasets Flickr30k and MSCOCO. As the threshold increases, there are fewer concepts that meet that minimum threshold. Concepts belonging in large numbers of images are therefore more common concepts.

In the Flickr30k dataset there are many object types, but the text annotations add extra complexity to the objects. Many objects are referred to with different names (*e.g.* car/vehicle/automobile). For general computer vision tasks, including object detection and localisation, localising a generic term for the object is satisfactory without the need to learn alternate words that describe the same object. Additionally, it would be favourable to learn a number of common concepts and avoid the larger number of rare concepts where there is minimal training data.

To perform localisation using no bounding box annotations we can perform unsupervised learning by leveraging the available full-text global image annotations that appear with each image in the dataset. Assuming that the global text annotations contain object-nouns that correspond to semantic objects in the image, the phrases can be broken down into their noun-phrases. We assume that the presence of the object-noun indicates the presence of that object in the image. Since spatial object localisation can be performed as a side-effect of object detection [80] these object-nouns will be the basis of finding the objects.

There are many object-nouns in the data. We use the off-the-shelf toolbox NLTK [13] to extract object-nouns from the available sentences on each image, after removing common stop-words. Looking at the Flickr30k dataset, there are many object nouns, and it may be unfitting to learn the presence of each type. There are 11,707 nouns detected, although many are counted despite being spelling errors, or non-object nouns. Instead of learning 11,707 possible objects, nouns are only used when they appear in a minimum number of images' annotations in the dataset. We choose the threshold of 50 images for training on both the MSCOCO train set and the Flickr30k train set.

Implementation

Visual pipeline. We construct the visual pipeline as in Figure 5.3 but consisting of a single convolutional object detector. The VGG_{16} classifier and softmax are removed so the output is from the Pool_5 layer. The model is trained on the new MSCOCO train set as in [7, 29, 44, 129], and also on Flickr30k train set for comparison. Model is validated using the Flick30k validation set. We also train separately with the ResNet_{152} model used by [29] for further comparison, and remove the final two layers of the ResNet (average pooling layer and softmax layer) to produce a FCN. The output of the ResNet_{152} is a tensor of size $2048 \times W \times H$.

Embedding space. We use the Google Model Word2Vec [69] with 300-D vectors as the high-dimensional word embedding space.

Training details. The convolution module Φ is trained from epoch zero, and finetuning of VGG_{16} begins from epoch 8. In finetuning the first four convolution layers are still frozen as they represent the detection of simple features.

The model is trained in batches of 64 image/phrase pairs using stochastic gradient descent with a learning rate of 0.01 and a momentum value of 0.9. Early experiments using Adam optimizer produced numerical instabilities, so stochastic gradient descent was preferred.

Preliminary experiments showed that the model training performance largely plateaus between epochs 50 and 70, so the model is trained for 80 epochs for confi-

dence. To account for dips in performance during the plateauing period, the offline stage is performed after each 10 epochs to determine model performance and the best model retained.

Experimental Results

Our VGG_{16}-only model when trained on MSCOCO produced a pointing game result of 55.59% and when trained on Flickr30k 58.67%. The results are shown in Table 5.2. On the more difficult task of training on MSCOCO train set (as compared to training with the Flickr30k train set) for validation on Flickr30k val set, the model outperforms the baselines Center, Random, and VGG_{16}-only. The result was also a 13.19% improvement over [129] and 5.49% better than [94]. While it was also better than [44], their model was trained instead on the Visual Genome [54] dataset.

We show some qualitative results in Figure 5.4 from our model trained on MSCOCO. In it we highlight some of the behaviours of the model. We see the model has some primitive ability to not only detect people, but in some cases differ between gender. In the first two images the model distinguishes between 'man' and 'girl'. In other cases the highest spatial activation point is slightly outside of the bounding box, representing a *miss* for that phrase, despite being close.

We can also observe that the model, while detecting the 'dog' class, focuses on only one portion of the dog, on the head. In another case it can detect a BMX bikes but chooses only to see the distant one. In the bottom-center image the model is apparently confused by the addition of the word 'yellow' and the neuronal activation for that object causes heatmaps on all yellow objects.

Method	Settings	Training	Flickr30k Val
Baseline	Center	-	49.20
	Random	-	27.24
	VGG_{16}[103]	-	35.37
	$ResNet_{152}$[103]	-	38.03
Javed [44]	VGG_{16}	VG	49.10
Zhang [129]	GoogleNet	MSCOCO[62]	42.40
Fang [31]	VGG_{16}	MSCOCO	29.03
Ramanishka [94]	InceptionV3[108]	MSVD[15], MSR-VTT[120]	50.10
Akbari [7]	ELMo+VGG_{16}	MSCOCO	61.66
Akbari [7]	ELMo+PNASNet	MSCOCO	69.19
Ours	VGG_{16}	MSCOCO	55.59
	VGG_{16}	Flickr30k	58.67
	$ResNet_{152}$	MSCOCO	59.04
	$ResNet_{152}$	Flickr30k	62.38

Table 5.2: Pointing game results for our model, and a series of results from the literature for comparison.

Figure 5.4: A selection of images from the Flickr30k validation set overlaid with trained localisation heatmaps for the text phrase written in the bottom left corners. The model was trained on MSCOCO using the VGG_{16} feature extractor. Areas of higher activation are red while lower activations are blue. The highest activation point used for the pointing game is represented as a white X. The bounding boxes for the localisation phrase are shown, and if the X is inside a box it is shown as green, otherwise as red. Best viewed digitally and in colour.

Effect of Differing Parallel Convolution Detectors

Our initial model was composed of a single convolution layer that acted as an object detector. Intuitively, the addition of multiple such layers in parallel can provide additional discriminative power to the model. Such multi-detectors have been used in [28] with spatial pooling for part detection.

We are motivated to observe the impact of erasure on a pair of parallel modules affects the performance of our model. In our first experiment we used a single convolution layer as an object detector as a baseline. In this experiment we will increase the number of parallel modules in our model.

Implementation

We construct the same model but with a series of parallel Φ convolution detectors. Each detector is independent and has its own set of trainable weights. The model is trained for 80 epochs using batches of 64 with stochastic gradient descent, with a learning rate of 0.01 and momentum 0.9. The Google Model Word2Vec [69] is used for the word embedding space.

The convolution module in the baseline model is a 1×1 convolution that outputs a tensor $\phi \in \mathbb{R}^{C \times W \times H}$ where W and H are the same spatial dimensions as the input tensor $\boldsymbol{X}$. This is because the learned 2-D kernel is a 1×1 kernel. Selecting kernel sizes of different spatial dimensions would produce output tensors that differ, and tensors $\phi_i..\phi_N$ require interpolation to a common size for heatmap generation. In these cases we use bilinear interpolation and resize the smaller outputs *upward* to match the spatial output size of the largest tensor. Intuitively, larger kernel sizes pool larger areas and would discriminate visual features of large size while smaller kernel sizes observe finer visual details, and while naturally a kernel size of $W \times H$ would act as a global pooling layer.

We train the model with a number of alternative module combinations and show some in Table 5.3.

Experimental Results

Preliminary experiments showed that two or three modules outperformed a single module, but performance enhancements peaked at three modules. Manual observation of the outputs of the modules showed that different modules produced identical output tensors, and the learned weights had been trained to look for the same visual features, as there was no constraints to make them learn complementary features.

Adding an additional parallel module increased pointing game performance in all cases, and we found that extra modules with kernel size of 1 was better at improving performance than larger kernel sizes. Adding one extra module with kernel size of 1 on the VGG_{16}-based model trained on MSCOCO increased the performance from 55.59% to 56.99% (1.4% improvement) and another addition further increased it to 57.60% (0.61% improvement), but further extra modules had negligible effect on performance.

We also show some results while training on Flickr30k - the VGG_{16}-based model with modules (1,2,4) in size increased performance against the baseline from 58.67% to 60.41%, which was a 1.74% increase in performance, and the ResNet_{152}-based model with modules (1,2,4) improved the performance similarly from 62.38% to 64.11%, marking an improvement of 1.73%.

Method	CNN Model	Training	Flickr30k Val
Baseline (1)	VGG_{16}	MSCOCO	55.59
	VGG_{16}	Flickr30k	58.67
	$ResNet_{152}$	MSCOCO	59.04
	$ResNet_{152}$	Flickr30k	62.38
(1,1)	VGG_{16}	MSCOCO	56.99
(1,4)	VGG_{16}		56.52
(1,1,1)	VGG_{16}		57.60
(1,2,8)	VGG_{16}		57.29
(1,1)	$ResNet_{152}$		59.08
(1,4)	$ResNet_{152}$		59.59
(1,2,4)	VGG_{16}	Flickr30k	60.41
(1,2,4)	$ResNet_{152}$		64.11

Table 5.3: Pointing game results from the baseline model and for differing number of parallel modules. Under the 'method' heading the kernel sizes are shown in parentheses. For example, (1,2) means two parallel convolution modules Φ_1 and Φ_2 with kernel sizes 1 and 2, respectively.

Complementary Learning With Erasure

Object part detection can using multiple detectors to discriminate more regions of the object [28]. Multi-feature detectors can be trained by masking the input image with the activated region of one detector, and re-feeding the image through the base model and a second object detector [117] to capture more features. Another approach is to use two detectors, but instead of masking the original image, mask the highest activation outputs of the first detector [132].

We modify our model to perform the erasure technique by replacing the parallel modules with two in series.

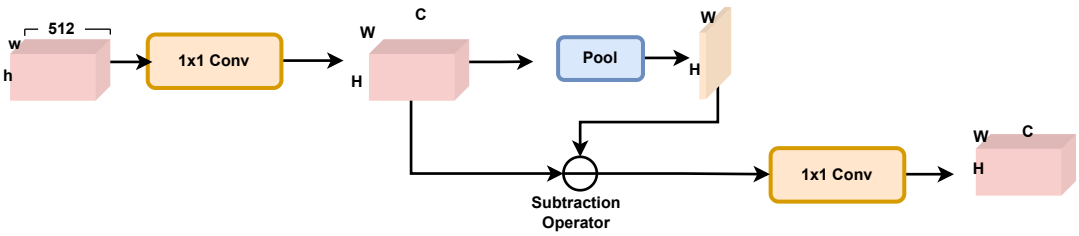

Figure 5.5: Block diagram of serial module to perform erasure. The first tensor $\boldsymbol{X} \in \mathbb{R}^{512 \times h \times w}$ that is outputted from the VGG_{16} feature extractor. The output of the first convolution layer is a tensor $\boldsymbol{Y} \in \mathbb{R}^{C \times H \times W}$, which is spatially pooled. The subtraction of the spatially pooled 2-D map from the tensor $\boldsymbol{Y}$ is fed into a second detector.

Implementation

We consider a set of two convolution layers with 1×1 kernels that act object detectors, and accept input tensors $\boldsymbol{X} \in \mathbb{R}^{C \times h \times w}$ where C is 512 for VGG_{16} or 2048 for ResNet_{152}. We re-implement the object detection stage as a serial rather than a parallel manner. The serial module is shown in Figure 5.5. The output is a tensor $\boldsymbol{X} \in \mathbb{R}^{C \times h \times w}$, and the output from the first convolution layer is a tensor $\boldsymbol{Y} \in \mathbb{R}^{C \times H \times W}$ that aims to perform the object detection as in the our base model. A pooling layer transforms tensor Y into a 2-D heatmap corresponding to the highest activations from the detector, which is then subtracted from the tensor $\boldsymbol{X}$ along the spatial dimension. This will mask the highest activations detected from the first convolution before feeding into the second to produce $\boldsymbol{Z} \in \mathbb{R}^{C \times H \times W}$, the final heatmap stack, and can be sliced channel-wise for object detection.

Preliminary experiments showed similar train time to the base model, so this model is trained for 80 epochs using batches of 64 with stochastic gradient descent, with a learning rate of 0.01 and momentum 0.9. The Google Model Word2Vec [69] is used for the word embedding space.

Experimental results

Method	CNN Model	Training	Flickr30k Val
Baseline (1)	VGG_{16}	MSCOCO	55.59
	VGG_{16}	Flickr30k	58.67
	$ResNet_{152}$	MSCOCO	59.04
	$ResNet_{152}$	Flickr30k	62.38
(1,1)	VGG_{16}	MSCOCO	56.99
(1,4)	VGG_{16}		56.52
(1,1,1)	VGG_{16}		57.60
(1,2,8)	VGG_{16}		57.29
(1,1)	$ResNet_{152}$		59.08
(1,4)	$ResNet_{152}$		59.59
(1,2,4)	VGG_{16}	Flickr30k	60.41
(1,2,4)	$ResNet_{152}$		64.11
Serial	VGG_{16}	MSCOCO	58.51
	$ResNet_{152}$		60.24

Table 5.4: Pointing game results including experiments from the baseline, the parallel modules, and the model with serial convolution layers with subtraction erasure. Under the 'method' heading the kernel sizes in the parallel modules are shown in parentheses. Kernel formats in parallel layers are as in Table 5.3. Modules used in the serial model are both 1×1 kernel sizes.

Both models trained on MSCOCO and using two convolution layers with the erasure method performed better than the model with two modules in parallel. The VGG_{16} model achieved a pointing game performance of 58.51%, which is 1.52% higher than two parallel modules. The $ResNet_{152}$ model trained similarly also outperformed the equivalent parallel module, with a pointing game performance of 60.24%, an improvement of 1.16%.

We observe in some cases, especially smaller objects, where the model is distracted

by some surrounding features as the erasure causes the primary feature to be masked and the surrounds detected as part of the object. A sample of heatmap-overlaid Flickr30k validation images are shown in Figure 5.6. The visualised heatmaps are the outputs from the model trained on MSCOCO (second-to-last column in Table 5.4). In Figure 5.6 the bottom-center image shows the phrase 'a meal' is to be localised on the image, and our model detects food items throughout the kitchen. The bottom-left image shows our model detecting both dogs but correctly identifies the black dog, as the word 'black' provides significant contextual cue. Likewise, the final image of the taxi in the city shows distraction by the model on the detection of yellow features, which was the case in the model using a single convolution object detectors as well.

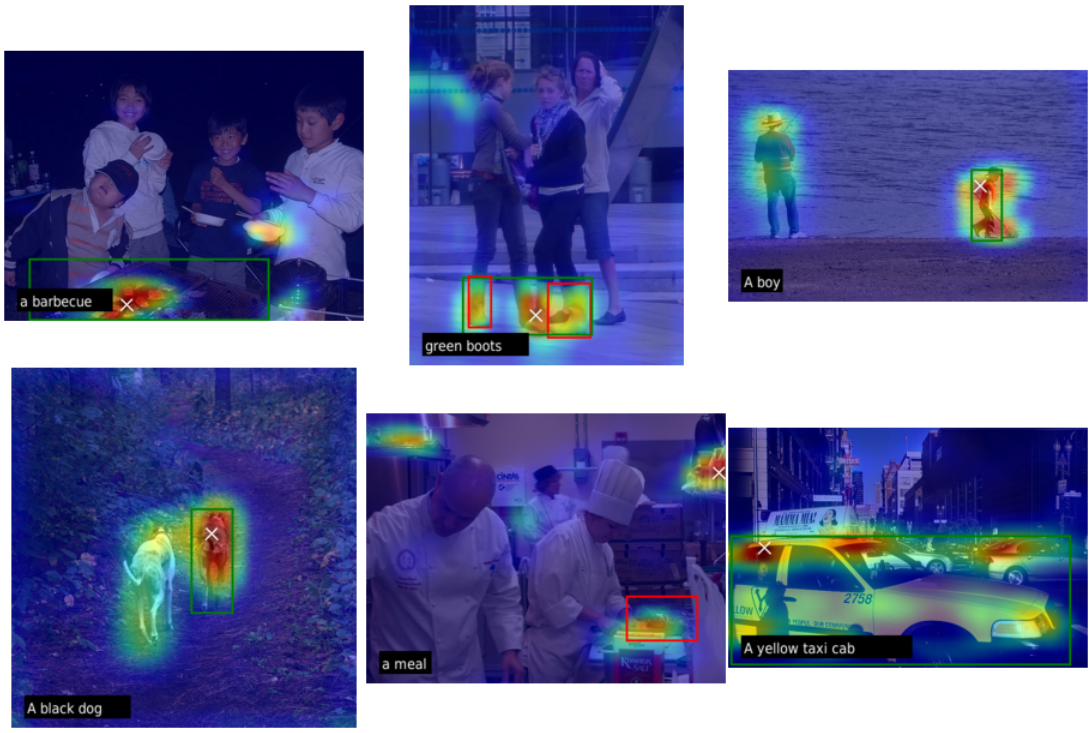

Figure 5.6: A set of six example Flickr30k validation images with overlaid heatmaps from the model trained with the erasure module and with the VGG_{16} feature detector on MSCOCO train set. Areas of higher activation are red while lower activations are blue. The highest activation point used for the pointing game is represented as a white X. The bounding boxes for the localisation phrase are shown, and if the X is inside a box it is shown as green, otherwise as red. Best viewed digitally and in colour.

Chapter Conclusion

Our model was designed to train on object localisation on a dataset of image-phrase pairs to localise visual objects from short English text phrases. We considered how the existence of nouns in the training phrases could be sufficient context for the existence of related visual objects in the paired images. Convolution layer-based object detection modules were able to detect semantic visual objects across the validation set, and we showed that uniting a number of modules is a simple and inexpensive way to boost localisation performance.

We chose a balance of object nouns to train a sufficient vocabulary while disregarding rare nouns, and utilised the semantic structure of an off-the-shelf Word2Vec model for the offline stage, suggesting such a pre-trained model can provide good semantic text information. Furthermore, replacement of parallel modules with two convolution layers in serial with simple erasure improved performance on localisation.

Chapter 6

Text Guided Archival Image Retrieval using Localisation Model

Introduction

The trained localisation model in the previous chapter used a pre-trained Word2Vec model for semantic structure in the offline stage and showed good promise for localising a set of trained vocabulary words, as well as unseen words by utilising the Word2Vec structure. Archival datasets such as NAA29k contain text metadata annotations paired to each image, where the text contains some contextual description of the image.

We are motivated to use the trained model of the previous chapter to attempt to enhance the image retrieval performance of NAA29k by utilising its capability to produce attention on visual features that correspond to text. We showed in the image retrieval chapter that improved global image pooling by focusing on the best features can improve retrieval performance. We desire to establish whether we can boost performance using the pretrained model on the dataset with text metadata as guidance to heatmap generation.

Implementation

The best model is used from the previous chapter trained on MSCOCO - the model with the serial module trained using erasure, and using the VGG_{16} feature extractor as used in the chapter on image retrieval.

The NAA29k dataset, 28,912 images, is fed into the model. The input size of each image is 256×256. Using the text phrases that are provided with each image, an output heatmap is generated and interpolated to the original input image size to produce a localisation for that phrase.

Visualising Localising Text on Archival Dataset

We provide some examples from the output of the trained localisation model. Figure 6.1 illustrates the model's ability to localise on particular objects as referenced in the text. The first image shows high activation on the lizard and the second image on the emus that are in the text annotations. The final image shows an airplane, and the model is able to localise the entirety of the object.

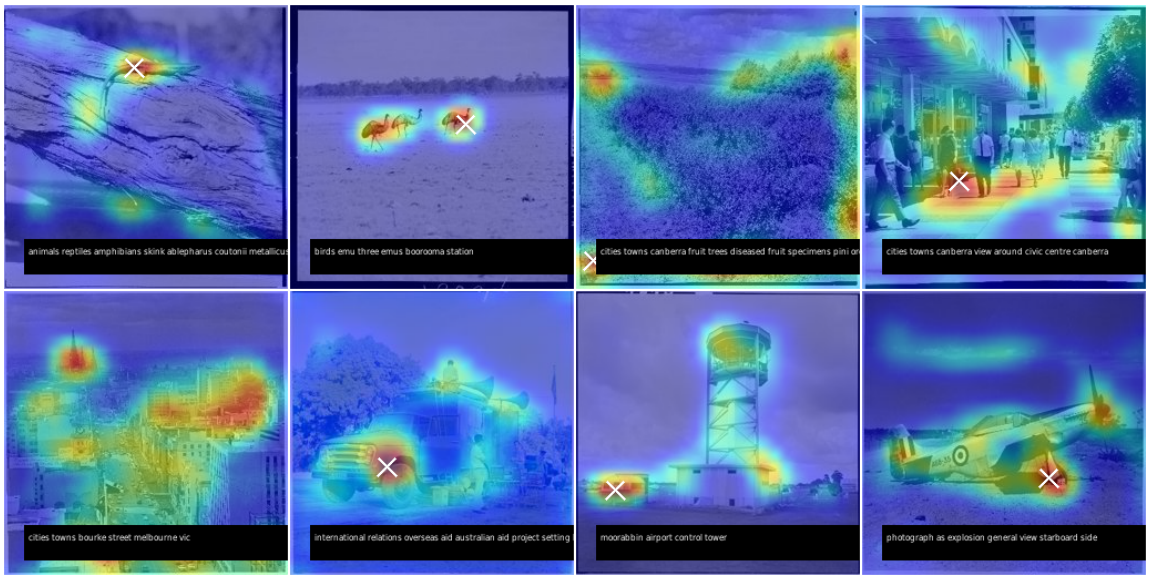

Figure 6.1: Eight images from the NAA29k dataset and their overlaid heatmaps generated from the text phrases. Higher neuronal activations are shown in red, and lower activations in blue. The most activated spatial location is shown with a white X. Phrase is embedded in black rectangles. Best viewed digitally and in colour.

The trained model when used on the NAA29k dataset shows some weaknesses, including behaviour that provides too much focus on a single visual feature and not enough on the entirety of the image. Recall that the NAA29k dataset is scene-heavy, and retrieval experiments showed that broader global information is more useful for pooling into global image descriptors than very specific parts. Figure 6.2 shows three such images where the model puts high activations on a single image portion and low on all others. The final image is of a telescope, and the phrase being 'research telescope', but the model does not see the dish portion of the object, only the building, when pooling using text guidance.

Figure 6.2: Three images from the NAA29k dataset and their overlaid heatmaps generated from the text phrases. These heatmaps suffered from excessive spatial focus by the model, and ignored more of the scene. Higher neuronal activations are shown in red, and lower activations in blue. The most activated spatial location is shown with a white X. Phrase is embedded in black rectangles. Best viewed digitally and in colour.

Furthermore, Figure 6.3 highlights how the attention can focus on salient parts of images. Six images from construction sites are shown and the common parts of semi-constructed piping receives attention, while other parts do not. The performance of each query depends on whether it is scene-based or object-based, and a balanced attention because the pooled descriptors would be influenced by the focus given by the text metadata. Focusing on only rare local features is an advantage for retrieval performance.

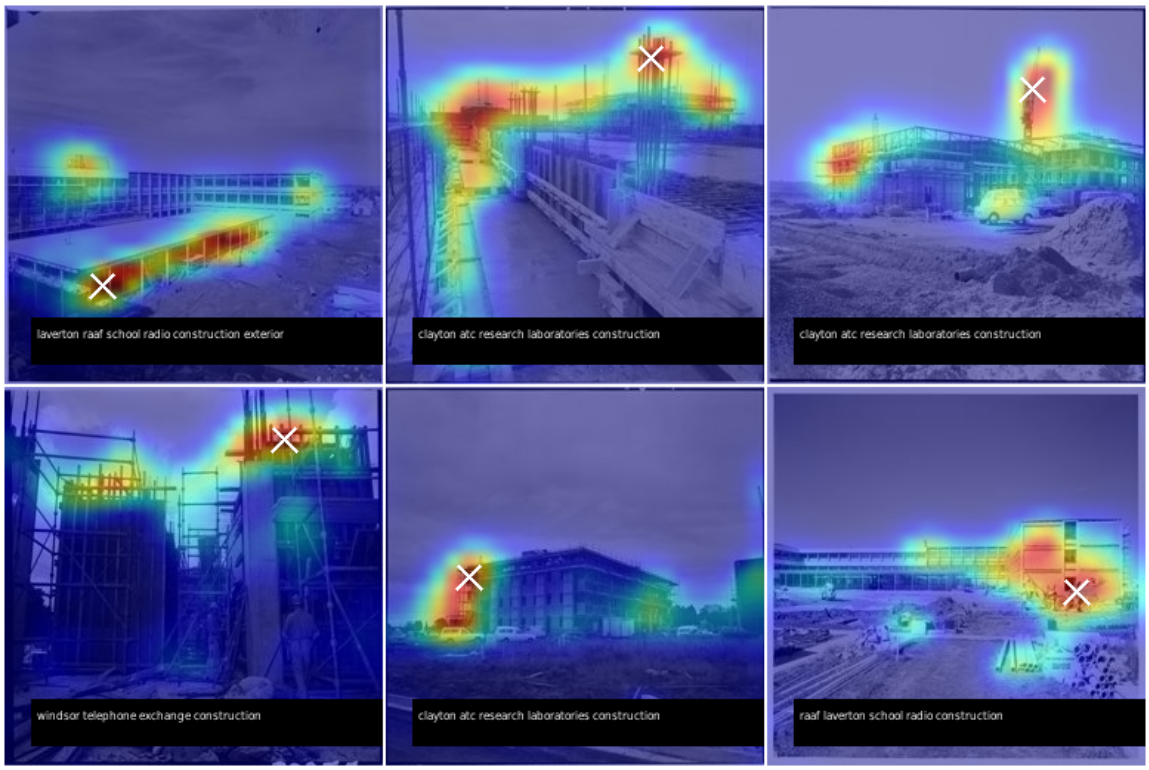

Figure 6.3: Six images from the NAA29k dataset from a construction site, and their overlaid heatmaps generated from the text phrases. Note the model focuses on common features in each. Higher neuronal activations are shown in red, and lower activations in blue. The most activated spatial location is shown with a white X. Phrase is embedded in black rectangles. Best viewed digitally and in colour.

Therefore, the weakness of this proposed model is its tendency to sometimes focus too heavily on some localised features and ignore global features, depending on the content of the provided text metadata. To balance this we also provide a new scalar weight hyperparameter β that can be chosen to weigh the heatmap tensor. Weighting is performed with a multiplication of $\boldsymbol{X} \times h \times \beta$ over the spatial dimension of $\boldsymbol{X}$ before the pooling stage. β is the same for the entire dataset and does not change for individual images. When the weight is zero the output is the same as without the weighting module. For our experiments we choose β as 0.0, 0.1, 0.5, and 1.0, reflecting varying levels of importance of the weighting module.

To perform retrieval using this proposed method we use the text-guided heatmaps

produced by the trained localisation model, weigh their values using β, and add to the output tensor of the pre-trained FCN portion of the VGG_{16} model. The VGG_{16} model accepts an input tensor $\boldsymbol{I} \in \mathbb{R}^{3\times w\times h}$ and outputs a tensor $\boldsymbol{X} \in \mathbb{R}^{512\times W\times H}$ where W and H are the width and height of the tensor. For our purposes we use the output of the Pool_5 layer, so when $w = 256$ and $h = 256$, $W = 8$ and $H = 8$ in the output tensor.

The output of the trained localisation model on image I_i is a predicted tensor $p_i \in \mathbb{R}^{8\times 8}$.

Retrieval Results

We show our retrieval results for NAA29k_{100} and NAA29k_{1137} in Table 6.1 using MaxPool and SumPool methods, and include $\beta = 0$ for completeness, which is equivalent to no guided text annotation. We find that using text guided heatmaps produce improved results.

For the NAA29k_{100} the weighting improved the MaxPool performance by 0.43 for $\beta = 0.1$, 0.92 for $\beta = 0.5$, and 0.33 for $\beta = 1.0$. Slightly better improvements were seen on the SumPooling method, which already presented comparatively higher mAP values to MaxPool. Using $\beta = 0.5$ produced the best result at 42.7, an increase of 1.2. We suspect that it is a balance between the model's ability to guide activations towards visual features mentioned in the text annotations, while helping to ignore non-visual nouns. This is a symptom of text annotations containing non-visual words and historical facts rather than literal visual descriptions of the scene. Take example of the two first images in Figure 6.1, where specific mentions of visual features ('lizard' and 'emu') are highly useful to the localisation model, when in other images more vague descriptions of the image and historical information are less useful to the text guided module.

For comparison we show fifteen query images and their top results on the baseline SumPooling method in Figure 6.4 and the same queries using text guidance in our proposed model in Figure 6.5.

It is clear that the performance improvement of the proposed model relies on valuable text metadata to accompany the images. Good metadata provides usable noun phrases for guided heatmap generation. To illustrate this, in Figure 6.6 we show the metadata of the queries in Figure 6.5 and Figure 6.4. The varying quality and comprehensiveness of the descriptions would affect the heatmap generation. This justifies our choice of introducing the β hyperparameter to soften the influence of the text guided module. However, for practical purposes, human intervention would be pragmatic to select an optimum β, with a higher value for datasets with visually-descriptive metadata.

Method	β	NAA29k$_{100}$	NAA29k$_{1137}$
MaxPool*	0.0	36.06	30.63
SumPool*	0.0	41.50	35.03
MaxPool	0.1	36.49	31.08
MaxPool	0.5	36.98	31.49
MaxPool	1.0	36.39	30.98
SumPool	0.1	42.48	36.10
SumPool	0.5	**42.70**	**36.18**
SumPool	1.0	41.96	35.63

Table 6.1: Mean Average Precision results for the NAA29k dataset on both groundtruths (100 and 1137) on MaxPool and SumPool methods using the VGG$_{16}$ model, with heatmaps generated from the trained localisation model. Methods indicated with an asterisk (*) are baseline methods reported in Table 4.3 for comparison.

Figure 6.4: Fifteen random queries from the NAA29k$_{100}$ groundtruth set are illustrated with their top-10 results on baseline SumPooling. The first image in each row is a query, and the following images are the top 10 results. Correct images have green borders and incorrect images have red borders. Best viewed digitally and in colour.

Figure 6.5: Fifteen queries from the NAA29k$_{100}$ groundtruth set from Figure 6.4 using text guidance and SumPooling with $\beta = 0.5$. The first image in each row is a query, and the following images are the top 10 results. Correct images have green borders and incorrect images have red borders. Best viewed digitally and in colour.

Crane store lighters progress shots and launching 234 11/14 July 1975

Cities and towns - Canberra, Civic Centre, Australian Capital Territory - Aerial view of Canberra, Civic Centre, the commercial and retail heart of the city in foreground

Tullamarine Airport - special extensions

Murwillumbah Post Office - [Item undated]

Garden Island dockyard - Sydney

North Sydney Post Office

Northern Territory - [Possibly Wycliffe Well alternative view - well built up on stone and concrete base. Pipes leading from well to large dark tank up on brick stand base. Ladder leaning against base. Sandy ground around]

Cities and towns - Melbourne - Aerial photograph of Melbourne

Inglewood Post Office

Tullamarine special extensions

Engineering - Heavy

States - NSW (Sydney) - Showgrounds - Easter show

Australian pastures may soon play an important part in the cattle industry of Nigeria. This is the opinion of Nigerian agriculturalist, Mr Eromosele, in Australia on a specialised four to six months course in pasture management under the Colombo Plan - Mr Eromosele pictured with Australian research scientist, Dr E F Henzell examining grasses at the Australian CSIRO [Commonwealth Scientific and Industrial Research Organisation], Pasture Research Station, Samford, near Brisbane, Queensland [photographic image]. 1 photographic negative: b&w, acetate

Laverton RAAF school of radio - interior

Casuarina office model

Figure 6.6: Fifteen queries as shown in Figure 6.4 and Figure 6.5 with their accompanying metadata. Best viewed digitally and in colour.

Chapter Conclusion

We used our proposed text guided localisation model to extract features from the archival dataset NAA29k and perform image retrieval. The pretrained model provided heatmap outputs that represent the confidence scores of visual features that correspond to the provided text annotations.

The quality of the provided metadata affects the final spatial weights, and some images and their text guided weights are illustrated. The weakness of the proposed model is that it requires strong and useful metadata to boost retrieval performance. Visualisations of the model's generated heatmaps showed some problems with images where over-focusing causes significant parts of the image to be ignored, while in other images the model is able to see large parts of the scene. We introduce a hyperparameter to balance the effect of the text guided weight module. We believe that the text guided module provides a worthy contribution to archival image retrieval where images are accompanied by text metadata as it provided a modest retrieval performance boost. Future work should concentrate on further investigating a text-guided model that minimises the distracting effect of non-visual text metadata, such as separate β values for individual images.

Chapter 7

Conclusion

Image retrieval on archival image datasets is an interesting visual task that can allow the public to easily access, browse, and explore historical digitised archives that have previously been in only analogue form. Being able to retrieve images from image queries can allow members of the public to easily access history and perform local research. In this thesis we explored pooling methods for image retrieval and how the methods affect the retrieval performance of several publicly accessible retrieval datasets and an archival set, NAA29k, from the National Archives of Australia.

In our literature review we established the strength of the pooling method and attention as essential to adequate image retrieval performance. Furthermore, the existence of accompanying text metadata motivates a strategy to harness it to boost performance.

We performed comprehensive experiments using Convolutional-Neural-Network-based feature extraction with different models, layer outputs, and pooling methods. We also implemented the diffusion process as an additional step to boost retrieval performance.

We proposed a simple change to the SPoC pooling method that harnesses the information within dataset galleries to provide a more useful pooling to boost performance. We introduce a weighting parameter that tweaks the gaussian weighting function, and we find it produces improved performance on the benchmark and

archival datasets.

In the localisation task we explore weakly-supervised training whereby a dual-path text/image model learns object localisation by utilising text object-nouns as training signals. We train a visual pipeline to detect object classes and examine the idea that an off-the-shelf Word2Vec model has sufficient semantic structure in place of a text model. The use of the Word2Vec model not only localised words in the known vocabulary, but the semantic structure provided an ability to localise for untrained words.

We proposed a further improvement on the localisation model by extending the object detector module with a set of parallel object detection modules, as well as using an erasure technique designed to find more object parts. The model performed better using a simple inclusion of more parallel modules. Furthermore, using the erasure technique in a serial manner produced further improvement over the parallel modules.

Considering the archival image dataset also contains text annotations, we were motivated to investigate if our trained localisation model can provide outputs to enhance the image retrieval performance. Our experiments showed that there is a small improvement, with qualitative examination showing that there are also distractions caused by the model focusing on small areas. We proposed a simple balance by introducing a hyperparameter that weights the effect of the text guidance. This demonstrated that guiding the weighting for pooling using text is a promising technique for image retrieval using archival digitised galleries with accompanying text annotations.

Image retrieval of digitised historical images is an appealing topic because of its practical applications in local research and democracy. We demonstrate good retrieval performance with CNN-based techniques and produce modest improvements with our proposed localisation model with text guidance. However, we identify a weakness of the proposed method and introduce a simple change to offer balance of the text guided weightings. There is strong motivation to continue using available text metadata in archival sets where available, to boost retrieval performance.

Suggestions for Future Research

We showed that retrieval on archival datasets can be improved by continuing to take advantage of the provided text annotations that accompany the digitised images. We considered object nouns as single-word phrases with the Word2Vec model embeddings. Further possible model improvements could include embedding whole text phrases, and adjusting the model's design to accept long phrases.

Our object nouns were limited by the use of an off-the-shelf object-noun detector. Many text phrases in the archival dataset contains specific information, such as names, places, and historical information, that does not correspond to direct visual objects. Further filtering of the annotations should help to focus on visual objects and provide less distractions from non-object nouns.

For localisation, utilising parallel object detectors boosted performance against a single object detector. Further, implementing erasure to detect additional features in a single, serial module further boosted performance. An interesting open question would be whether a parallel set of such modules could further improve localisation. Further using the trained model as the feature extractor, how it could improve retrieval performance on the archival dataset.

Our proposed model used a single parameter to balance the effect of the text guided output against the visual-only output. However, it is a single parameter that affects all images equally. An interesting future direction would be to explore how the parameter can be chosen for individual image/text pairs according to the text metadata content. Improvement here would significantly reduce the constraint of the proposed model.

Bibliography

[1] Bibliothèque Nationale de France. URL: http://images.bnf.fr, Accessed: 28 September 2018.

[2] Bodlean Ballads Search. URL: http://zeus.robots.ox.ac.uk/ballads/page0, Accessed: 28 September 2018.

[3] Collection Online. URL: http://www.britishmuseum.org/research/collection_online/search.aspx, Accessed: 27 September 2018.

[4] Histórico do Arquivo Público Mineiro: instituição cultural mais antiga de minas gerais. URL: http://www.siaapm.cultura.mg.gov.br/modules/wfchannel/index.php, Accessed: 27 September 2018.

[5] PhotoSearch. URL: https://recordsearch.naa.gov.au/SearchNRetrieve /Interface/SearchScreens/PhotoSearch.aspx, Accessed: 27 September 2018.

[6] (2013). Facebook Users Are Uploading 350 Million New Photos Each Day. Insider Inc, URL: https://www.businessinsider.com/facebook-350-million-photos-each-day-2013-9, Accessed: 17 Oct 2018.

[7] Akbari, H., Karaman, S., Bhargava, S., Chen, B., Vondrick, C., and Chang, S. (2018). Multi-level multimodal common semantic space for image-phrase grounding. *CoRR*, abs/1811.11683.

[8] Arandjelovic, R. and Zisserman, A. (2013). All about vlad. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 1578–1585, Washington, DC, USA. IEEE Computer Society.

[9] Babenko, A. and Lempitsky, V. S. (2015). Aggregating deep convolutional features for image retrieval. *Computing Research Repository*, abs/1510.07493.

[10] Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V. S. (2014). Neural codes for image retrieval. *Computing Research Repository*, abs/1404.1777.

[11] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 404–417, Berlin, Heidelberg. Springer-Verlag.

[12] Bergamo, A., Bazzani, L., Anguelov, D., and Torresani, L. (2014). Self-taught object localization with deep networks. *Computing Research Respository*, abs/1409.3964.

[13] Bird, Steven, E. L. and Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.

[14] Chen, C.-c., Wactlar, H. D., Wang, J. Z., and Kiernan, K. (2005). Digital imagery for significant cultural and historical materials. *International Journal on Digital Libraries*, 5(4):275–286.

[15] Chen, D. and Dolan, W. (2011). Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA. Association for Computational Linguistics.

[16] Chen, K., Kovvuri, R., Gao, J., and Nevatia, R. (2017). Msrc: Multimodal spatial regression with semantic context for phrase grounding. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, page 23–31, New York, NY, USA. Association for Computing Machinery.

[17] Chung, J. S., Arandjelović, R., Bergel, G., Franklin, A., and Zisserman, A. (2015). Re-presentations of art collections. In Agapito, L., Bronstein, M. M., and Rother, C., editors, *Computer Vision - ECCV 2014 Workshops*, pages 85–100, Cham. Springer International Publishing.

[18] Cimpoi, M., Maji, S., Kokkinos, I., and Vedaldi, A. (2015). Deep filter banks for texture recognition, description, and segmentation.

[19] Clough, P., Sanderson, M., and Reid, N. (2003). The eurovision st andrews photgraphic collection (esta).

[20] Crowley, E. J. and Zisserman, A. (2014). In search of art. In *Workshop on Computer Vision for Art Analysis, ECCV*.

[21] Csillaghy, A. and Benz, A. (1999). Interactive image retrieval in large astronomical archives: The aspect system. *Solar Physics*, 188:203–216.

[22] Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

[23] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA. IEEE Computer Society.

[24] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[25] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[26] Dogan, P., Sigal, L., and Gross, M. H. (2019). Neural sequential phrase grounding (seqground). *CoRR*, abs/1903.07669.

[27] Donoser, M. and Bischof, H. (2013). Diffusion processes for retrieval revisited. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1320–1327.

[28] Durand, T., Mordan, T., Thome, N., and Cord, M. (2017). Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5957–5966.

[29] Engilberge, M., Chevallier, L., Pérez, P., and Cord, M. (2018). Finding beans in burgers: Deep semantic-visual embedding with localization. *CoRR*, abs/1804.01720.

[30] Enser, P. (1995). Progress in documentation pictorial information retrieval. *Journal of Documentation*, 51:126–170.

[31] Fang, H., Gupta, S., Iandola, F. N., Srivastava, R. K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J. C., Zitnick, C. L., and Zweig, G. (2014). From captions to visual concepts and back. *CoRR*, abs/1411.4952.

[32] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., and Mikolov, T. (2013a). Devise: A deep visual-semantic embedding model. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc.

[33] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., and Mikolov, T. (2013b). Devise: A deep visual-semantic embedding model. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[34] Gevers, T. and Smeulders, A. W. M. (2000). Pictoseek: combining color and

shape invariant features for image retrieval. *IEEE Transactions on Image Processing*, 9(1):102–119.

[35] Gong, Y., Wang, L., Guo, R., and Lazebnik, S. (2014). Multi-scale orderless pooling of deep convolutional activation features. *Computing Research Repository*, abs/1403.1840.

[36] Gonzales, R., W. R. and Eddins, S. (2004). *Digital Image Processing With Matlab*. Prentice-Hall.

[37] He, K., Zhang, X., Ren, S., and Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *Computing Research Repository*, abs/1406.4729.

[38] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

[39] He, Y., Zhang, X., and Sun, J. (2017). Channel pruning for accelerating very deep neural networks. *Computing Research Repository*, abs/1707.06168.

[40] Hirata, K. and Kato, T. (1992). Query by visual example - content based image retrieval. In *Proceedings of the 3rd International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '92, pages 56–71, London, UK, UK. Springer-Verlag.

[41] Hoang, T., Do, T., Tan, D. L., and Cheung, N. (2017). Selective deep convolutional features for image retrieval. *Computing Research Repository*, abs/1707.00809.

[42] Hu, J., Shen, L., and Sun, G. (2017). Squeeze-and-excitation networks. *Computing Research Repository*, abs/1709.01507.

[43] Jain, A. K. and Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233 – 1244.

[44] Javed, S. A., Saxena, S., and Gandhi, V. (2018). Learning unsupervised visual grounding through semantic self-supervision. *CoRR*, abs/1803.06506.

[45] Jayaraman, D. and Grauman, K. (2017). Learning to look around: Intelligently exploring unseen environments for unknown tasks. *Computing Research Repository*, abs/1709.00507.

[46] Jegou, H., Douze, M., and Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 304–317, Berlin, Heidelberg. Springer-Verlag.

[47] Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *CVPR 2010 - 23rd IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, San Francisco, United States. IEEE Computer Society.

[48] Kalantidis, Y., Mellina, C., and Osindero, S. (2015). Cross-dimensional weighting for aggregated deep convolutional features. *Computing Research Repository*, abs/1512.04065.

[49] Karpathy, A. and Fei-Fei, L. (2014). Deep visual-semantic alignments for generating image descriptions. *CoRR*, abs/1412.2306.

[50] Kazemzadeh, S., Ordonez, V., Matten, M., and Berg, T. (2014). ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar. Association for Computational Linguistics.

[51] Kekre, D. H. B., Thepade, S. D., and Maloo, A. (2010a). Query by image content using colour averaging techniques. In *International Journal of Engineering Science and Technology*, 6, pages 1612–1622.

[52] Kekre, H., D. Thepade, S., and Akshay, M. (2010b). Query by image content using color-texture features extracted from haar wavelet pyramid. *International Journal of Computer Applications*, CASCT.

[53] Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539.

[54] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D. A., Bernstein, M. S., and Fei-Fei, L. (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR*, abs/1602.07332.

[55] Lam, D., Kuzma, R., McGee, K., Dooley, S., Laielli, M., Klaric, M., Bulatov, Y., and McCord, B. (2018). xview: Objects in context in overhead imagery.

[56] Larochelle, H. and Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order boltzmann machine. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 1243–1251. Curran Associates, Inc.

[57] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178.

[58] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324.

[59] Lei, T., Zhang, Y., and Artzi, Y. (2017). Training rnns as fast as cnns. *CoRR*, abs/1709.02755.

[60] Lewis, P. H., Martinez, K., Abas, F. S., Fauzi, M. F. A., Chan, S. C. Y., Addis, M. J., Boniface, M. J., Grimwood, P., Stevenson, A., Lahanier, C., and Stevenson,

J. (2004). An integrated content and metadata based retrieval system for art. *IEEE Transactions on Image Processing*, 13(3):302–313.

[61] Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *Computing Research Repository*, abs/1312.4400.

[62] Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

[63] Liu, J. (2013). Image retrieval based on bag-of-words model. *Computing Research Repository*, abs/1304.5168.

[64] Liu, L., Shen, C., and v. d. Hengel, A. (2017). Cross-convolutional-layer pooling for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2305–2313.

[65] Liu, Y., Wang, K., Li, Q., He, R., Yuan, Y., and Zhang, H. (2020). Weakly supervised arrhythmia detection based on deep convolutional neural network. *CoRR*, abs/2012.05641.

[66] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157.

[67] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

[68] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[69] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q.,

editors, *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc.

[70] Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

[71] Mohedano, E., McGuinness, K., Giró i Nieto, X., and O'Connor, N. E. (2017). Saliency weighted convolutional features for instance search. *Computing Research Repository*, abs/1711.10795.

[72] Mohedano, E., McGuinness, K., O'Connor, N. E., Salvador, A., Marques, F., and Giro-i Nieto, X. (2016). Bags of local convolutional features for scalable instance search. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ICMR '16, pages 327–331, New York, NY, USA. ACM.

[73] Mojsilovic, A., Kovacevic, J., Hu, J., Safranek, R. J., and Ganapathy, S. K. (2000). Matching and retrieval based on the vocabulary and grammar of color patterns. *IEEE Transactions on Image Processing*, 9(1):38–54.

[74] Mopuri, K. R. and Babu, R. V. (2015). Object level deep feature pooling for compact image representation. *Computing Research Repository*, abs/1504.06591.

[75] Mousavian, A. and Kosecka, J. (2015). Deep convolutional features for image based retrieval and scene categorization. *Computing Research Respository*, abs/1509.06033.

[76] Murray, N. and Perronnin, F. (2014). Generalized max pooling. *Computing Research Repository*, abs/1406.0312.

[77] Nakka, K. K. and Salzmann, M. (2018). Deep attentional structured representation learning for visual recognition. *Computing Research Repository*, abs/1805.05389.

[78] Ng, J. Y., Yang, F., and Davis, L. S. (2015). Exploiting local features from deep networks for image retrieval. *Computing Research Repository*, abs/1504.05133.

[79] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2161–2168, Washington, DC, USA. IEEE Computer Society.

[80] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2015). Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 685–694.

[81] Owais, M., Arsalan, M., Choi, J., and Park, K. R. (2019). Effective diagnosis and treatment through content-based medical image retrieval (cbmir) by using artificial intelligence. *Journal of Clinical Medicine*, 8(4):462.

[82] Pang, S., Ma, J., Xue, J., Zhu, J., and Ordonez, V. (2018). Deep feature aggregation with heat diffusion for image retrieval. *Computing Research Repository*, abs/1805.08587.

[83] Pass, G. and Zabih, R. (1999). Comparing images using joint histograms. *Multimedia Systems*, 7(3):234–240.

[84] Pass, G., Zabih, R., and Miller, J. (1996). Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM International Conference on Multimedia*, MULTIMEDIA '96, pages 65–73, New York, NY, USA. ACM.

[85] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E.,

and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[86] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[87] Pereira, A. M. Bastos, D. R. L. M. and Li, F. (2019). Multilayer convolutional feature aggregation algorithm for image retrieval. *Mathematical Problems in Engineering*, 10.1155/2019/9794202.

[88] Petrelli, D. and Clough, P. (2012). Analysing user's queries for cross-language image retrieval from digital library collections. *The Electronic Library*, 30(2):197–219.

[89] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[90] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[91] Picard, D., Gosselin, P., and Gaspard, M. (2015). Challenges in content-based image indexing of cultural heritage collections. *IEEE Signal Processing Magazine*, 32(4):95–102.

[92] Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *CoRR*, abs/1505.04870.

[93] Radenovic, F., Tolias, G., and Chum, O. (2017). Fine-tuning CNN image retrieval with no human annotation. *Computing Research Repository*, abs/1711.02512.

[94] Ramanishka, V., Das, A., Zhang, J., and Saenko, K. (2016). Top-down visual saliency guided by captions. *CoRR*, abs/1612.07360.

[95] Razavian, A. S., Sullivan, J., Maki, A., and Carlsson, S. (2014). A baseline for visual instance retrieval with deep convolutional networks. *Computing Research Repository*, abs/1412.6574.

[96] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

[97] Reid, N. H. (1999). The photographic collections in st andrews university library. *Scottish Archives*, 5:83–90.

[98] Sanchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, 105(3):222–245.

[99] Schaefer, G. and Stich, M. (2004). Ucid - an uncompressed colour image database. In *In Storage and Retrieval Methods and Applications for Multimedia 2004, volume 5307 of Proceedings of SPIE*, pages 472–480.

[100] Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):530–535.

[101] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681.

[102] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and Lecun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. *http://arxiv.org/abs/1312.6229*.

[103] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Computing Research Repository*, abs/1409.1556.

[104] Singh, K. K. and Lee, Y. J. (2017). Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. *CoRR*, abs/1704.04232.

[105] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380.

[106] Sukhia, K. N., Riaz, M. M., Ghafoor, A., and Ali, S. S. (2020). Content-based remote sensing image retrieval using multi-scale local ternary pattern. *Digital Signal Processing*, 104:102765.

[107] Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1):11–32.

[108] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567.

[109] Takami, M., Bell, P., and Ommer, B. (2014). An approach to large scale interactive retrieval of cultural heritage. In *Eurographics Workshop on Graphics and Cultural Heritage*. The Eurographics Association, The Eurographics Association.

[110] Tolias, G., Sicre, R., and Jégou, H. (2015). Particular object retrieval with integral max-pooling of CNN activations. *Computing Research Repository*, abs/1511.05879.

[111] Tsai, C. (2007). A review of image retrieval methods for digital cultural heritage resources. *Online Information Review*, 31(2):185–198.

[112] Valle, E., Cord, M., and Philipp-Foliguet, S. (2006). Content-based retrieval of images for cultural institutions using local descriptors. In *Geometric Modeling and Imaging–New Trends (GMAI'06)*, pages 177–182.

[113] van de Sande, K., Gevers, T., and Snoek, C. (2010). Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596.

[114] Vardazaryan, A., Mutter, D., Marescaux, J., and Padoy, N. (2018). Weakly-supervised learning for tool localization in laparoscopic videos. *CoRR*, abs/1806.05573.

[115] Wang, L., Li, Y., and Lazebnik, S. (2017a). Learning two-branch neural networks for image-text matching tasks. *CoRR*, abs/1704.03470.

[116] Wang, P., Liu, L., Shen, C., Huang, Z., v. d. Hengel, A., and Shen, H. T. (2017b). Multi-attention network for one shot learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6212–6220.

[117] Wei, Y., Feng, J., Liang, X., Cheng, M., Zhao, Y., and Yan, S. (2017). Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. *CoRR*, abs/1703.08448.

[118] Woo, S., Park, J., Lee, J.-Y., and Kweon, I.-S. (2018). Cbam: Convolutional block attention module. *Computing Research Repository*, abs/1807.06521.

[119] Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., and Zhang, Z. (2014). The application of two-level attention models in deep convolutional neural network for fine-grained image classification. *Computing Research Repository*, abs/1411.6447.

[120] Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296.

[121] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *Computing Research Repository*, abs/1502.03044.

[122] Yang, S., Kim, Y., Kim, Y., and Kim, C. (2019a). Combinational class activation maps for weakly supervised object localization. *CoRR*, abs/1910.05518.

[123] Yang, S., Kim, Y., Kim, Y., and Kim, C. (2019b). Combinational class activation maps for weakly supervised object localization. *CoRR*, abs/1910.05518.

[124] Yang, W., Huang, H., Zhang, Z., Chen, X., Huang, K., and Zhang, S. (2019c). Towards rich feature discovery with class activation maps augmentation for person re-identification. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1389–1398.

[125] Yarlagadda, P., Monroy, A., Carque, B., and Ommer, B. (2011). Recognition and analysis of objects in medieval images. In Koch, R. and Huang, F., editors, *Computer Vision – ACCV 2010 Workshops*, pages 296–305, Berlin, Heidelberg. Springer Berlin Heidelberg.

[126] yuan Wang, X., feng Chen, Z., and jiao Yun, J. (2012). An effective method for color image retrieval based on texture. *Computer Standards & Interfaces*, 34(1):31 – 35.

[127] Yue, J., Li, Z., Liu, L., and Fu, Z. (2011). Content-based image retrieval using color and texture fused features. *Mathematical and Computer Modelling*, 54(3):1121 – 1127. Mathematical and Computer Modeling in agriculture (CCTA 2010).

[128] Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional networks. *Computing Research Repository*, abs/1311.2901.

[129] Zhang, J., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. (2016). Top-down neural attention by excitation backprop. *CoRR*, abs/1608.00507.

[130] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018a). The unreasonable effectiveness of deep features as a perceptual metric. *Computing Research Repository*, abs/1801.03924.

[131] Zhang, X., Wang, T., Qi, J., Lu, H., and Wang, G. (2018b). Progressive attention guided recurrent network for salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[132] Zhang, X., Wei, Y., Feng, J., Yang, Y., and Huang, T. S. (2018c). Adversarial complementary learning for weakly supervised object localization. *CoRR*, abs/1804.06962.

[133] Zhang, Y., Qian, X., and Tan, X. (2015). Sketch-based image retrieval using contour segments. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6.

[134] Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. (2014). Object detectors emerge in deep scene cnns. *Computing Research Repository*, abs/1412.6856.

[135] Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. (2015). Learning deep features for discriminative localization. *Computing Research Repository*, abs/1512.04150.

[136] Zhou, W., Li, H., and Tian, Q. (2017). Recent advance in content-based image retrieval: A literature survey. *Computing Research Repository*, abs/1706.06064.