UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Visual Search: Finding Similar Images

Author:
Görkem Çamlı

Supervisor:
Prof. Marta Arias Vicente

Master Thesis
Defense Date: 28 January 2020

MASTER IN INNOVATION AND RESEARCH IN INFORMATICS
DATA SCIENCE
FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)
UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) – BarcelonaTech

Abstract

Visual Search is an area that started to significantly grow and change the current search landscape both in the academy and industry. Visual Search task focuses on finding visually similar images given a query image and returning the results in a ranked order where the most similar images ranked first in the result. Although plenty of research and improvement is done in this area, the visual search concept is still an active research area and it is challenging to accomplish its goal.

The main contributions of this thesis are (1) implementing an end-to-end system to perform the visual search that can be used in further research or applications, and (2) conducting experiments on different types of feature extraction and dimensionality reduction methods to understand which ones are more likely to give better search relevance and quality results.

Currently, there are different approaches used to create visual search systems one of these approaches uses directly deep learning with networks such as Siamese or Triplet Networks to learn similarity metrics. The other approach is using pre-trained Networks to extract features and apply search algorithms on top of the features to find similar images. The end-to-end system created for this thesis uses the second approach. The module notion is used and different steps of visual search implemented in different modules with the idea of having each step independent from each other. This is especially important for debugging and having more focused research on specific steps. The Visual Search steps include feature extraction, dimensionality reduction, search and metric evaluation (results) modules. Each Experiment can also be integrated as different modules within the system. The end-to-end system is ready to use with any dataset and different modules can be added/removed very easily from the system, that is why it makes it advantageous to use the system in different research groups for different research questions in visual search area yet having a stable system to compare.

In addition to the visual search system developed, 3 Experiments are conducted to answer different research questions on visual search. First Experiment focuses on answering "Which image feature representation methods are better in the context of visual search according to the defined evaluation metrics?". As feature extraction methods both simple and complex methods are used such as color histogram, color moments, histogram of gradients, CNN feature vector embeddings extracted from the last fully connected layer of the pre-trained CNNs with ImageNet data. Experiment 2 takes a different direction from Experiment 1 and aims to answer "Combining features extracted from different feature extraction method can perform better than only using one method?", this is to understand if the features from different extraction methods can be complementary to each other rather than being alternatives. Hence the features used in the experiment 1 are normalized and merged with different feature combinations and compared

if the results are quantitatively and qualitatively improved. The last Experiment answers which dimensionality reduction method should be preferred, if any, in the context of visual search. In the third experiment, different unsupervised and supervised dimensionality reduction methods are used such as PCA, UMAP, Random Projections, LDA and NCA.

To evaluate the system and the experiments different evaluation metrics are used, these metrics can be gathered under 3 main areas: Category Prediction metrics, Search Relevance Metrics and Efficiency Metrics. Category Prediction metrics assess how many of the retrieved results for a given query image has the same class and superclass of the query image. The Search Relevance calculates how ranking and retrieved list of a retrieved result matches with the baseline results. The Efficiency measures the index and search times for different models.

The first experiment's result shows that CNN features perform better than simple and hand-crafted features, however not all of the CNN networks perform well on visual search. The best performing feature is found the ones that are extracted from the InceptionResnetV2 pretrained network. In Experiment 2, we see that different types of features can complement each other rather than being chosen as alternatives to each other. Combining basic features and combining different CNN features improves both evaluation metrics and qualitative results but it hurts the efficiency of the models since it increases the search time. The increase in the search time, in the 3rd experiment, proved to be improved by using accuracy-time trade-off in the visual search by applying dimensionality reduction.

Contents

Contents iii

1 Introduction 1
- 1.1 Motivation . . . 2
- 1.2 Challenges . . . 2
- 1.3 Goals and Research Questions . . . 3
- 1.4 Thesis Outline . . . 3

2 State of the Art 4
- 2.1 Image Retrieval with Deep CNN Features . . . 4
- 2.2 Siamese Network and Triplet Network . . . 5
- 2.3 End-to-End Visual Search Systems developed by Companies . 7
  - 2.3.1 Pinterest . . . 7
  - 2.3.2 eBay . . . 7
  - 2.3.3 Others . . . 8

3 Methodology & Development of Work 9
- 3.1 Modules . . . 9
  - 3.1.1 Feature Extraction Modules . . . 10
  - 3.1.2 Dimensionality Reduction Module . . . 13
  - 3.1.3 Visualization Module . . . 15
  - 3.1.4 Search Module . . . 16
  - 3.1.5 Results and Experiments Module . . . 16
- 3.2 Frameworks . . . 17
  - 3.2.1 Baseline Framework . . . 17
  - 3.2.2 Model Framework . . . 18

4 Data 19
- 4.1 Data Exploration . . . 20

5 Evaluation 21

5.1 Category Prediction Metrics 21
5.1.1 Class@10 21
5.1.2 SuperClass@10 22
5.2 Search Relevance Metrics 22
5.2.1 Intersection@10 22
5.2.2 AP@10 and MAP@10 23
5.3 Efficiency Metrics 23
5.3.1 Index and Search Time 24
5.4 Letter-Value Plots 24

6 Experiments & Results **25**
6.1 Experiment 1 - Baselines 25
6.1.1 Experiment 1 Summary 25
6.1.2 Experiment 1 Results 25
6.2 Experiment 2 - Baselines: Basic Features vs Combined Features 32
6.2.1 Experiment 2 Summary 32
6.2.2 Experiment 2 Results 33
6.3 Experiment 3 - Baseline vs Models 39
6.3.1 Experiment 3 Summary 39
6.3.2 Projections 41
6.3.3 Experiment 3 Results 45

7 Conclusion & Future Works **53**

A Appendix **56**
A.1 Experiment 1 Additional Figures and Tables 57
A.2 Experiment 2 Additional Figures and Tables 62
A.3 Experiment 3 Additional Figures and Tables 66

Bibliography **71**

Chapter 1

Introduction

Visual Search or Content-based Image Retrieval [9] is a growing field of research both in academia and in industry. With the increase of uploading online unstructured data such as image, the ways of finding, using and sharing these unstructured contents have changed by users. The availability of the unstructured data increased the demand to browse and search them freely, and lead the way to the visual search where the similar images can be searched with other images.

The focus of the visual search task is ”Given an image query, retrieving a ranked list of visually similar images to the image query”. The common techniques used for visual search are creating Siamese or Triplet networks or creating an end-to-end system with the combination of different steps such as feature extraction, indexing, and retrieval.

Even though many technological and research advances happened lately, Visual Search is still an ongoing research field with many challenges and problems to solve.

This thesis provides an end-to-end Visual Search framework, with different modules where the visual search process is implemented step by step. Within the framework, it is possible to create different Visual Search models with different feature extraction methods and dimensionality reduction methods to improve efficiency later in the search and retrieval step.

Therefore, the main contributions of this thesis are to provide an end-to-end framework performing visual search task and the results of the experiments done to understand the performances of different feature extraction and dimensionality reduction methods in a visual search context.

The end-to-end framework can be easily modified and extended with different modules. This framework can be used as a starting point and used with any image dataset by other researchers to further advance this research area.

1.1 Motivation

Until recently, it was very difficult to do visual search and get relevant results. Therefore, we mostly relied on textual descriptions to perform image search. With the advances in computer vision and deep learning, now it is possible to perform visual search.

The possibility of searching images with another image also make it possible to decrease the language barriers exist in traditional image search. Instead of image search with text, searching images via an image query provides the users an opportunity to search for similar patterns, designs or shapes they don't know how to describe properly with text.

In addition to the decrease in language barriers, understanding visual search and its characteristics can be used in different image application areas such as finding near-duplicate image detection, image recommendations, different product recommendations such as fashion synthesis [21].

Another motivation is that visual search is already started to take an important role in our daily lives with many retailer companies and with different applications such as Google Lens and Pinterest Lens. Hence, understanding and contributing to this area is not only applicable but its results can be seen in daily lives.

Therefore, the motivation to make a research on the topic of Visual Search comes from the above three reasons: (1) the advances in computer vision and deep learning opened new possibilities to perform visual search and thus improved search experience by decreasing language barriers, (2) visual search can potentially affect and improve different application areas in image-related tasks and (3) advances in visual search can have a direct impact on daily lives.

1.2 Challenges

Similarity measure and image representation are critical for the Visual Search task. Although different approaches are being used, it remains as a current field of study to find a good similarity measure and image representation that is accepted and widely used. Below are some of the challenges of visual search task:

- How do we represent images?
- What do we mean by image similarity and how should we define it? How we should measure it?
- How we can do image retrieval based on semantic information of the image semantic than basic features?

Within the scope of this thesis, we will focus on the image representation part and its final effect on the visual search task.

1.3 Goals and Research Questions

The research questions we want to answer for this thesis are:

- Which image feature representation method is better in the context of visual search?
- Can combining different feature representations increase the relevance and category prediction results of the visual search?
- Should one dimensionality reduction method be preferred over another in the context of visual search?

1.4 Thesis Outline

This thesis report starts with an explanation of the state-of-art methods and models and systems created by companies such as Pinterest. In Chapter 3, we explain the methodology and visual search framework we created for this thesis. Chapter 4 covers the information about the data used during the thesis and experiments. Chapter 5 explains the evaluation metrics we decided to use for categorization, search relevance and efficiency. Chapter 6 shares the experiments and their results and concludes with the interpretation of the results. We conclude the thesis with Chapter 7 explaining the conclusions we have gathered and possible future works that could be done.

Chapter 2

State of the Art

State of the Art section explains commonly used methods and state of the art techniques for content-based image retrieval. A brief explanation of Image Retrieval with Deep CNN Features, Siamese Network, and Triplet Network is shared to give more context about the recent work in this area. At the end of this section, the end-to-end visual search systems developed by tech companies are explained to show how companies use these techniques in a real-life context and scale it to millions of users.

2.1 Image Retrieval with Deep CNN Features

With the recent improvements in the deep learning area, deep CNNs perform very well on visual tasks such as object detection and classification. It is also shown that deep neural network architectures can capture semantic information [19] and outperform the other handcrafted image descriptor techniques. Hence, one of the most commonly used techniques is to leverage the performance of the CNN models and use their feature mappings containing semantic abstractions that are close to human perception [12, 10] in the image retrieval task.

Visual Search is then created by having a content-based image retrieval pipeline containing different modules. Within the Visual Search context these pipelines generally divided into two or three main modules: feature extraction and feature matching modules [28] or feature extraction, query, and retrieval modules. The feature extraction module is for creating the feature vector (image representation) from a given raw input image. The features could be any of the color, histogram, shape, texture, CNN features or image transform features that could represent the image. Query module is where the query image's features are extracted in a way that it becomes ready to be compared with the other image representations in the database. Retrieval module compares the distances between the query image repre-

sentation and the other images' representations in the database, finds the images that have the least distance to the query image, ranks the results and returns the visually most similar images to the query image. In some systems query and retrieval, the module is combined together as a feature matching module.

To create the image representations state-of-the-art models are using CNN models to create image embeddings. The embeddings can be created either with a pre-trained, fine-tuned or a CNN model built/trained from scratch. Canziani compares the pre-trained networks submitted for ImageNet Challenges over the years to compare their accuracy vs operation times to see real-life practicability of these networks [5].

2.2 Siamese Network and Triplet Network

Another technique is to make models learn a general similarity function. This is rather learning to compare tasks than a classification task, where models learn feature similarity. Convolutional Neural Networks such as Siamese Network [23] and Triplet Network [29] are developed to learn to compare images and find similar images by optimizing feature distance of the image pairs. The main task to achieve in both of these network types is to map similar images closer to each other whereas mapping dissimilar images away from each other [23].

Siamese Network

To learn a similarity function, Siamese Network uses a pair of images to feed the network. Figure 2.1, shows the Siamese Network structure where the model takes a pair of images as an input to 2 identical CNN branches sharing the same weights and parameters [23]. The pairs of images are either positive or negative pairs. Positive pairs are when the given 2 images are either on the same category or visually similar to each other whereas the negative pairs are not in the same category or visually dissimilar images.

The output of the CNN branches is tied to a loss layer that uses a contrastive loss. Loss function tries to minimize the distance between the CNN branch outputs of the positive image pairs and it tries to maximize the loss for the output of negative image pairs [23].

Triplet Network

Triplet Network [29], which is known as the current state-of-the-art method for image retrieval [13] is proposed by Want et al. is a deep ranking model where a triplet of images is taken as an input. A triplet consists of a query image called anchor, a positive image that is similar to the anchor and a negative image that is dissimilar to the anchor. Similar to the Siamese network these 3 images are fed to 3 identical CNNs with the same parameter and

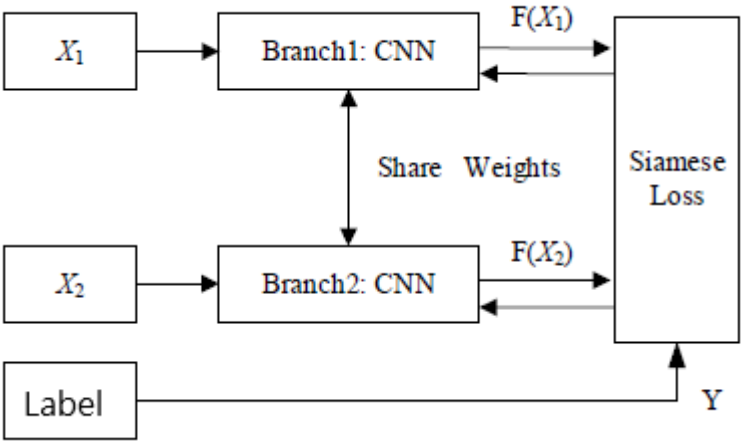

Figure 2.1: Siamese Network Structure [32]

weights during the training as seen in Figure 2.2. At the end of a training pass, each CNN branch outputs an embedding to represent each image.

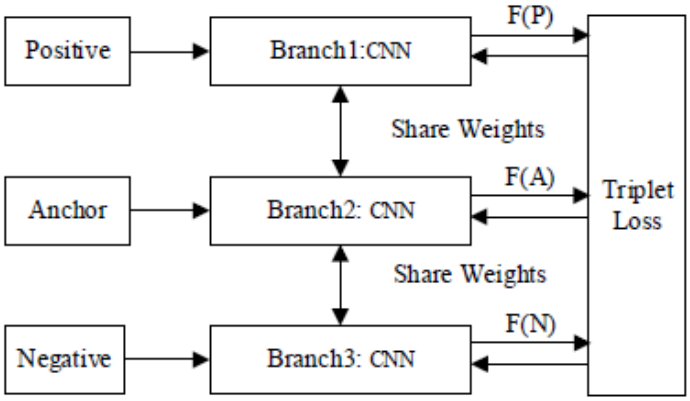

Figure 2.2: Triplet Network Structure [32]

The idea again is similar to the Siamese network, which is to train the network in a way that creates embeddings where the anchor and positive image embeddings similar/close to each other than the anchor and negative image embedding. To achieve that the pairwise distance between the anchor and positive image embeddings and pairwise distance between the anchor and negative image embeddings are found to calculate hinge loss. The goal is to minimize the hinge loss.

One of the main challenges for Siamese and Triplet Networks is the dataset

availability. A very common way to create the dataset is with classification datasets were putting the same classes as similar and different classes as dissimilar images, even though this is a work-around to create a dataset, it is biased towards class and objects in the model and the visual similarity of the images doesn't take into account except they are being in the same classes.

2.3 End-to-End Visual Search Systems developed by Companies

Companies such as Pinterest, Google (Lens), Amazon, eBay, and Alibaba have dedicated research teams working specifically on visual search and discovery [16, 31, 35]. Their published work explains the scalable end-to-end models they are created and the challenges they have within their product domain.

2.3.1 Pinterest

Pinterest is one of the companies that is pioneering the research in the visual search area [16, 33, 34]. They currently have 3 visual search and discovery services that serve for different aims: Flashlight, Shop-The-Look, and Lens [34]. Flashlight and Lens provides recommendation based on the selected image on Pinterest or taken by a camera. Shop-The-Look is a visual search service where users can search for similar products to the chosen query product.

Recently, they have shown that they are able to achieve good performance by unifying the image embeddings for these different services such as visual search and visual browsing/recommendation where they created a network on multi-task metric learning [34]. With the multi-task metric learning context, the created network tries to optimize on multiple similarity metrics to create a single unified embedding that performs well in different similarity contexts such as browsing similar images and searching for relevant products [34].

2.3.2 eBay

At eBay, the visual search engine has two main parts, Visual Index Ingestion to do category prediction and creating image hashes and Image Ranking which takes the query image category prediction and hash to match with the remaining of the products to retrieve similar product results [31]. To create the hash parts, they modify ResNet-50 in a way that they use a shared layer with 2 different output layers one making category recognition and the other creating the binary hash extraction [31].

The main challenges of creating a visual search engine for retail companies such as eBay is to be able to cope with the volatile inventory, scale the system, and provide consistent results even though with the bad data quality or bad quality of query image [31].

2.3.3 Others

Other recent works from big tech companies research teams are as follows:

Alibaba created a unified deep ranking framework using triplet networks to feed their visual search model with triplet images [35].

Amazon published a paper on multi-scale Siamese Network [1] showing that the created network captures better the fine-grained image similarities than a currently used traditional CNNs.

At Microsoft Bing, the visual search framework is first transforming the images as feature vectors where they use "Visual Words" and then they use a three-level cascaded ranker framework to retrieve similar images [15]. The Bag of Visual Words is created by feature extraction of deep neural networks for both image database and the query image. Then, feature vectors of image database are represented as Bag of Visual words, clustered and n-nearest centroid method is used for query feature vector to find the similar candidate images [15].

In Jet, they created a Visual Search Engine within Elastic Search: their approach is once they created their feature vectors, they encode them as strings and index both the feature vector and encoded strings to the Elasticsearch. At the search time, they retrieve the similar products based on the similarity of the encoded strings and rerank them based on the Euclidean Distance of the retrieve similar candidates [24].

As it can be seen both in industry and in academia, different techniques are being used for visual search. Image representation, similarity metric learning, and image matching are still active research areas within the visual search context.

Chapter 3

Methodology & Development of Work

The project is implemented in a way that given the configuration settings and a dataset, visual search can be performed with an end-to-end approach, where desired baseline and model frameworks are created including the evaluation of the created models.

In order to simplify the specific model types and the steps to create them, we introduce the module and framework notion below.

3.1 Modules

With the idea of code reusability and flexibility, we separated each step of the visual search into different modules and created our baseline and model framework by using these modules.

Each module can be run independently and have a specific role to accomplish.

Modules implemented within the scope of this thesis are:

- Feature Extractions
  - CNN Feature Extraction Module
  - Color Feature Extraction Module
- Dimensionality Reduction Module
- Visualization Module
- Search Module
- Results and Experiments Module

Using modules gives us the flexibility to create different specific frameworks as well as remove or add any additional modules if necessary with a minimum effort.

3.1.1 Feature Extraction Modules

As stated earlier one of the challenges of visual tasks is the representation of an image. What Feature Extraction module does is, it reads the image files within a given path, extracts the feature vector (image representation) based on the specified feature extraction type and saves the extracted feature vectors under the features folder.

We used different image representations: color, local descriptor and deep features of images to understand if any feature type is better than others and if combining them improves the quantitative and qualitative visual search results and experience.

CNN Feature Extraction Module

Deep features are the features we extracted from pre-trained convolutional neural networks (CNN) on image classification tasks. We used state-of-the-art CNN features that are widely known and used for image representations.

To extract the features, the top layer is removed and the penultimate layer which is the last fully connected layer (dense layer) is used in all of the networks. Imagenet weights are used and average pooling is applied to get the feature vectors. Table 3.1 shows the models used and the dimensions of the feature vector extracted per image.

CNN Architecture	*Dimensions of Feature Vector*
Inception V3	2048
VGG16	512
VGG19	1000
ResNet50	1000
InceptionResNetV2	1536
MobileNet	1024
Xception	2048

Table 3.1: CNN Architecture used for Deep Feature Extractions

Color Feature Extraction Module

Color features are basic features to describe images. For color features, we extracted Color Histogram and Color Moments of every image. These two feature types explain the color similarity between images. RGB color space is used.

Color Moments

4 color moments are extracted for each color channel. Color Moments extracted are mean, variance, skewness, and kurtosis of each color channel. Since the CIFAR-100 dataset contains an image with colors, each image has 3 channels. Therefore, the color moments feature gives a 12-dimension feature vector per image. The moments are calculated and defined as following [17, 20]:

p_{ij} is the value at image p's ith color channel and jth pixel. N is the total number of pixels on the given image.

Mean Moment: Mean Moment is the average color value in an image per channel.

$$E_i = \sum_{N}^{j=1} \frac{1}{N} p_{ij}$$

Variance Moment: Variance Moment is a descriptor of distribution of the pixel values per channel, it tells how far the values spread out from each other.

$$\sigma_i^2 = \frac{1}{N} \sum_{N}^{j=1} (p_{ij} - E_i)^2$$

Skewness Moment: Skewness Moment is the descriptor for the degree of asymmetry in the color distributions.

$$s_i = \sqrt[3]{\frac{1}{N} \sum_{N}^{j=1} (p_{ij} - E_i)^3}$$

Kurtosis Moment: Kurtosis Moment is to understand the shape of the probability distribution.

$$k_i = \sqrt[4]{\frac{1}{N} \sum_{N}^{j=1} (p_{ij} - E_i)^4}$$

Color Histogram

Color Histogram is used to have a better capture of the distribution of colors by image over the color space, hence to understand better the tone of the image compared to color moments.

Color Histograms are extracted per color channel, as bins of 100. Therefore per image, a total of 300 dimensions is returned as the color histogram feature vector.

The disadvantage of these two color features is that they only represent the color distribution but they disregard the location of the colors.

Histogram of Oriented Gradient

Histogram of Oriented Gradient (HOG) is a feature descriptor similar to the SIFT descriptors, edge orientation histograms and shape context [6].

The computation of HOG feature is created in five steps: image normalization (optional), computing the magnitude and direction of gradient of image applying the $[-101]$ filter in x and y-axis, computing histograms of gradients for every cell, normalizing the results across blocks and flattening the result into a feature vector [4].

A gradient's magnitude calculated as:

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}$$

A gradient's direction calculated as:

$$\theta = \tan^{-1}[\frac{g_y}{g_x}]$$

An illustration of HOG can be seen on Figure 3.1:

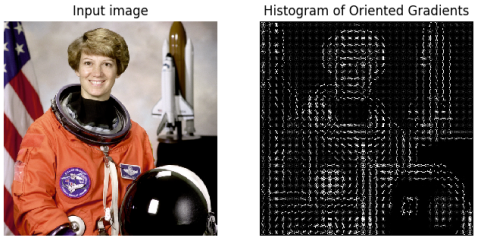

Figure 3.1: Image and it's Histogram of Gradients Visualization [4]

In this thesis, to extract HOG features, the histograms are calculated based on 8 orientations: these orientations can be represented as angles:
$\begin{bmatrix} 0 & 45 & 90 & 135 & 180 & 225 & 270 & 315 \end{bmatrix}$, each cell has 16x16 pixels, and each block has one cell.

3.1.2 Dimensionality Reduction Module

Dimensionality Reduction module is created to apply different dimensionality reduction techniques to the given raw feature vectors. Dimensionality Reduction module can only be used in the model creation, not on the baseline since baselines are by definition created with the raw extracted features.

Dimensionality Reduction module can be run with the dimensionality reduction type and the number of the desired dimension to reduce parameters. Dimensionality reduction methods are provided for this thesis:

Unsupervised Dimensionality Reduction Techniques

Principal Component Analysis, Uniform Manifold Approximation and Projection and Random Projections with Johnson-Lindenstrauss Lemma is used as unsupervised dimensionality reduction methods.

Principal Component Analysis

Principal Component Analysis (PCA) [25], is one of the most widely used dimensionality reduction methods. PCA is a linear dimensionality reduction method and for the projection of the data to reduce the dimension, it uses Singular Value Decomposition [30]. After the data projection, principal components are returned as the new feature dimensions, where each component is orthogonal to each other. The principal components are structured in a way that the first returned (smallest) components capture the largest possible variance of the data presented.

In the code, while applying PCA the given feature data is centered but not scaled.

Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP) [22], can be both used as a manifold learning technique for non-linear dimensionality reduction method and as for visualization. UMAP is based on Riemannian Geometry and algebraic topology [22].

UMAP algorithm has several phases: constructing a fuzzy topological representation and optimizing the representation of a low dimension. While computing the fuzzy topological representation the UMAP algorithm uses the Nearest-Neighbor-Descent algorithm of Dong et al. to find the local neighbors and the remaining of the computations are done for only on the found local neighbors, making the algorithm very efficient [22].

Johnson-Lindenstrauss Lemma and Random Projections

Johnson-Lindenstrauss Lemma states that it is possible to obtain a lower-dimensional Euclidean space while nearly preserving the pairwise distances

by applying random projections [3] on any datasets with high dimensions [8].

According to Johnson-Lindenstrauss Lemma, the reduced data is not dependent on the original data dimensions but rather depends on the number of samples.

Applying Johnson-Lindenstrauss Lemma is straightforward, matrix multiplication of the high-dimensional data and matrix M of shape $(d \times n)$, where M is constructed with entries of independent Gaussian random variables. n is the number of samples, and d is the minimum boundary dimension found by Johnson-Lindenstrauss Lemma. Each random variable r in the matrix M, $i = 1, ..., d$, let $r_i \in \mathbb{R}^n$, are created by entries independently and randomly drawn from Gaussian Distribution with mean 0 and variance 1, N(0,1) [8, 7].

The minimum number of components to guarantee the eps-embedding, d, is given by [8, 4]:

$$\text{number of components} >= \frac{4log(\text{number of samples})}{\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3}}$$

where epsilon is the distortion rate. The lower epsilon is the less distortion rate, meaning the pairwise distances preserved more accurately.

Figure 3.2 and Table 3.2 show the Johnson-Lindenstrauss Bounds for the size of the CIFAR 100 dataset which is 60K samples. 60K is taking as sample size instead of 50K (only training samples) because later in the query time, the pairwise distances for query images also will be calculated.

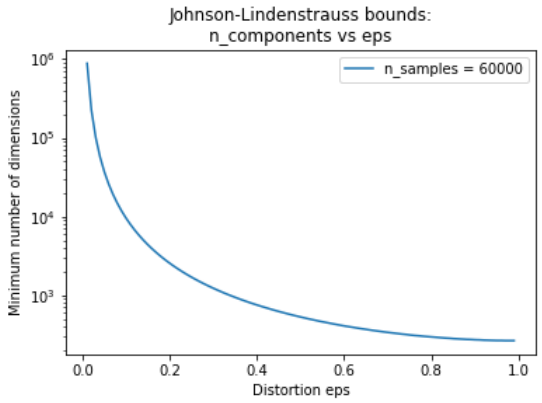

Figure 3.2: Johnson-Lindenstrauss Bounds for 60K samples [4]

Epsilon Range	Min n Components
0.01	886075
0.12	6763
0.23	2000
0.34	1001
0.45	630
0.55	454
0.66	358
0.77	304
0.88	274
0.99	264

Table 3.2: Calculated Epsilon ranges vs Min N Components required for 60K samples

Supervised Dimensionality Reduction Techniques

For Supervised dimensionality reduction techniques Linear Discriminant Analysis and Neighborhood Component Analysis are used:

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) [2], is a supervised method that finds linear decision boundaries between the classes of the input data. It can also be used as a dimensionality reduction method by projecting the data on the most discriminative directions found where these directions maximize the separation of the classes. In this way, the data is reduced to a linear subspace.

Neighborhood Component Analysis

Neighborhood Component Analysis (NCA) [11], is a non-parametric supervised method used both in metric learning and dimensionality reduction. NCA learns the distance, by applying linear transformations to the data by using supervised Leave-One-Out (LOO) classification. The linear transformations aim to maximize the performance of LOO, hence stochastic nearest neighbors rule in the projected space. NCA doesn't make assumptions on class distribution shapes and their boundaries [11]. According to the experiments done by Goldberger et al. it performs better than LDA and PCA in separating classes of the data [11].

During the execution of this module, the index time is recorded.

3.1.3 Visualization Module

Visualization Module is implemented to understand visually the space of the created feature vectors in 2 or 3 dimensions. Since created feature vectors' dimensions are much bigger than 2 or 3 dimensions, first dimensionality reduction is applied to be able to plot the feature vectors. By plotting the images embeddings we can have a sense of how good the feature extraction method is to separate different classes or images from each other. However, it is important to consider that the visualizations are only in 2-3 dimensions hence isn't directly representing the feature vector spaces but rather it is showing the feature vector spaces in smaller dimensions.

To run properly, the module needs a path to the feature vector, two-dimensionality reduction type to visualize and their dimensions. If nothing is specified by default PCA and t-SNE visualizations with 2 dimensions are created.

The module creates different plots of the given dataset: colored-point plot of embeddings (colors are coded based on labels), label plots of embeddings, side by side plot of PCA and t-SNE plots of the dataset in 2D or 3D. The plots are saved under the output folder. In Experiments chapter under Ex-

periment 3 section, PCA, UMAP and t-SNE visualizations of the CIFAR-100 dataset is shared.

3.1.4 Search Module

Search Module applies K-Nearest-Neighbor (KNN) search to the given feature vector(s) to find similar image candidates to the given query image.

Search Module completes below four objectives:

1. Find the nearest neighbors of a given input feature vector.
2. Shows and saves the image examples of found candidate images of a query image.
3. Write the results of the found nearest neighbor's in a file.
4. During the execution of the KNN search, the search time is recorded. It saves the search time result in a general measurement file.

The search module has 3 parameters: metric, number of neighbors and algorithm. By Default, the metric is euclidean, the number of neighbors, K is 10 and the algorithm is brute force. These parameters can be changed, K can be given as an integer number, the metric could be any of the distance metrics implemented in sklearn.neighbors.DistanceMetric or Pairwise metrics [4] such as euclidean, hamming, Jaccard, cosine similarity, etc, and algorithm could be anything between brute force, KDTree, BallTree, or auto (the algorithm is selected automatically based on data). For the Nearest Neighbor implementation sci-kit-learn library is used [4].

The main drawback of the KNN algorithm is that it becomes very expensive and slow to run the algorithm once the dataset gets bigger. For Visual Search sometimes approximate search techniques are more useful and efficient. Since our main research goals are focused on understanding the feature vectors and optimal dimensionality reduction techniques for them, we used an exact search approach. Our aim is to have exact distance results and rankings based on these distance results to be able to fully compare different feature vectors. Otherwise, using KNN wouldn't be very feasible nor practical in a real-life application due to its time complexity for brute-force KNN is $O(n \times m)$ where n is the number of training examples and m is the dimensions in the training set [26].

3.1.5 Results and Experiments Module

Results Module is the module executed at the end, after creating baseline and models. Result modules aim to compare models with each other and to their corresponding baseline based on the evaluation metrics explained in the Evaluation chapter. There are 3 main evaluation aspects: category

prediction, search relevance and efficiency. For each of these aspects, the corresponding measurements are calculated and a general description tables created by each model and baseline type along with diamond and distribution plots ready to be analyzed. These tables and plots are created both in model level, superclass level for a given model and label level for a given model to understand the results in high and low levels.

The experiments module is a more refined version of the Results module where a specific set of baselines and models are outlined and results are created for the outlined experiment.

3.2 Frameworks

Frameworks are end to end models created for visual search. For this thesis, 2 types of the framework created: baseline and model frameworks.

3.2.1 Baseline Framework

Baseline framework, shown in Figure 3.3, is the setting where the feature vectors are directly used for candidate search. Baseline Framework uses the feature extraction modules, visualization module (optional) and search module.

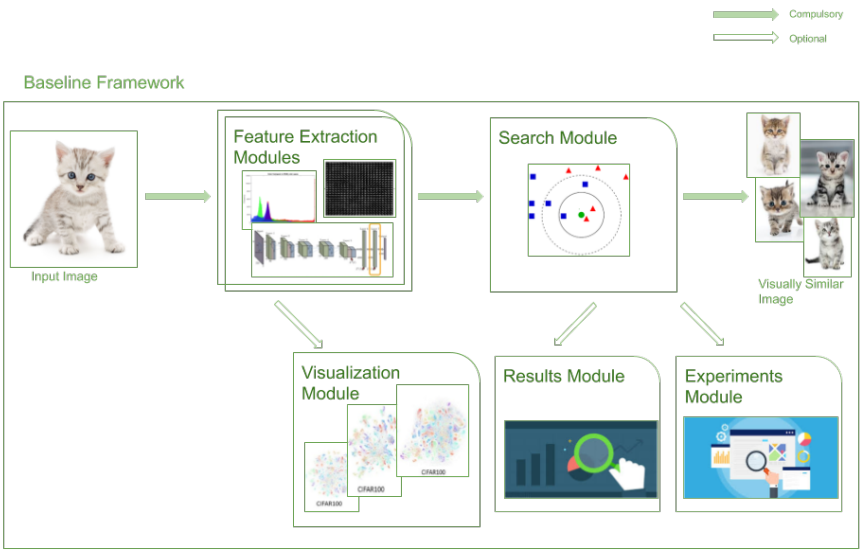

Figure 3.3: Baseline Framework

3.2.2 Model Framework

In the model framework, shown in Figure 3.4, before doing the search for similar candidates, dimensionality reduction is applied to the feature vector. Model Framework uses the feature extraction modules, dimensionality reduction module, visualization module (optional) and the search module.

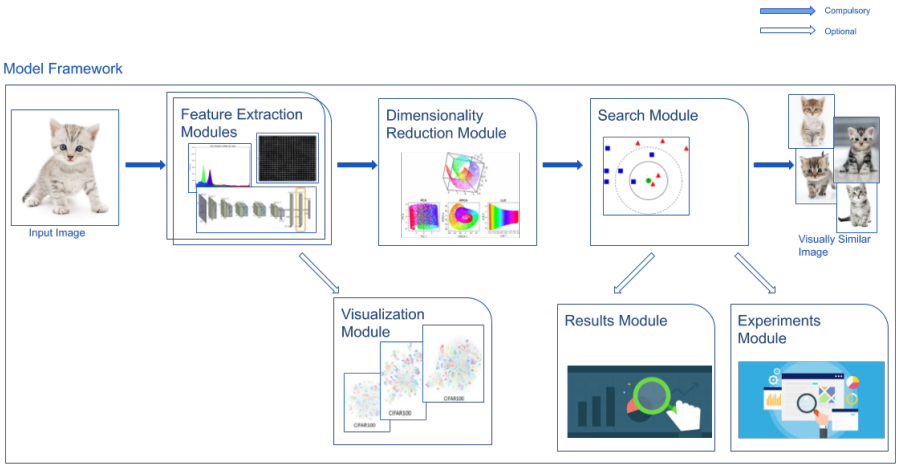

Figure 3.4: Model Framework

Hence, from now on, we will call the Baseline to the model where there is no dimensionality reduction module and model when dimensionality reduction applied during the visual search process.

Chapter 4

Data

Visual Search task is different than object recognition or classification task. However, there is no go-to dataset widely available for visual search. Visual Search datasets used by companies are not publicly available, or the ones that are available don't provide the true retrieval results to be able to do the evaluation.

That is why for the evaluation purposes of this thesis we used an object recognition dataset called CIFAR-100 [18]. CIFAR-100 dataset contains 100 classes where each class has 600 images (500 train and 100 test)- These 100 classes are distributed equally to 20 superclasses where each superclass have 5 classes. Dataset consists of 60,000 images each with 32x32 color images.

The superclass and the classes of the dataset can be seen from Table 4.1:

Superclass	*Classes*
Aquatic mammals	beaver, dolphin, otter, seal, whale
Fish	aquarium fish, flatfish, ray, shark, trout
Flowers	orchids, poppies, roses, sunflowers, tulips
Food Containers	bottles, bowls, cans, cups, plates
Fruit and Vegetables	apples, mushrooms, oranges, pears, sweet peppers
Household Electrical Devices	clock, computer keyboard, lamp, telephone, television
Household Furniture	bed, chair, couch, table, wardrobe
Insects	bee, beetle, butterfly, caterpillar, cockroach
Large Carnivores	bear, leopard, lion, tiger, wolf
Large Man-made Outdoor Things	bridge, castle, house, road, skyscraper
Large Natural Outdoor Scenes	cloud, forest, mountain, plain, sea
Large Omnivores and Herbivores	camel, cattle, chimpanzee, elephant, kangaroo
Medium-sized Mammals	fox, porcupine, possum, raccoon, skunk
Non-insect Invertebrates	crab, lobster, snail, spider, worm
People	baby, boy, girl, man, woman
Reptiles	crocodile, dinosaur, lizard, snake, turtle
Small Mammals	hamster, mouse, rabbit, shrew, squirrel
Trees	maple, oak, palm, pine, willow
Vehicles 1	bicycle, bus, motorcycle, pickup truck, train
Vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

Table 4.1: SuperClass and their Classes in the CIFAR 100 dataset

For the thesis and the implemented visual search framework the train images used as catalog, image database and test images used as query images.

Why this dataset? This dataset is selected for several reasons: the main reason is while creating the experiments and the system results we want to have a dataset where real-life disadvantages exist such as bad image quality, very diverse classes and images with complicated backgrounds. This was one of the big datasets but small enough to accomplish the research experiments given limited resources.

4.1 Data Exploration

Figure 4.1 shows a random image of each class in the CIFAR-100 dataset.

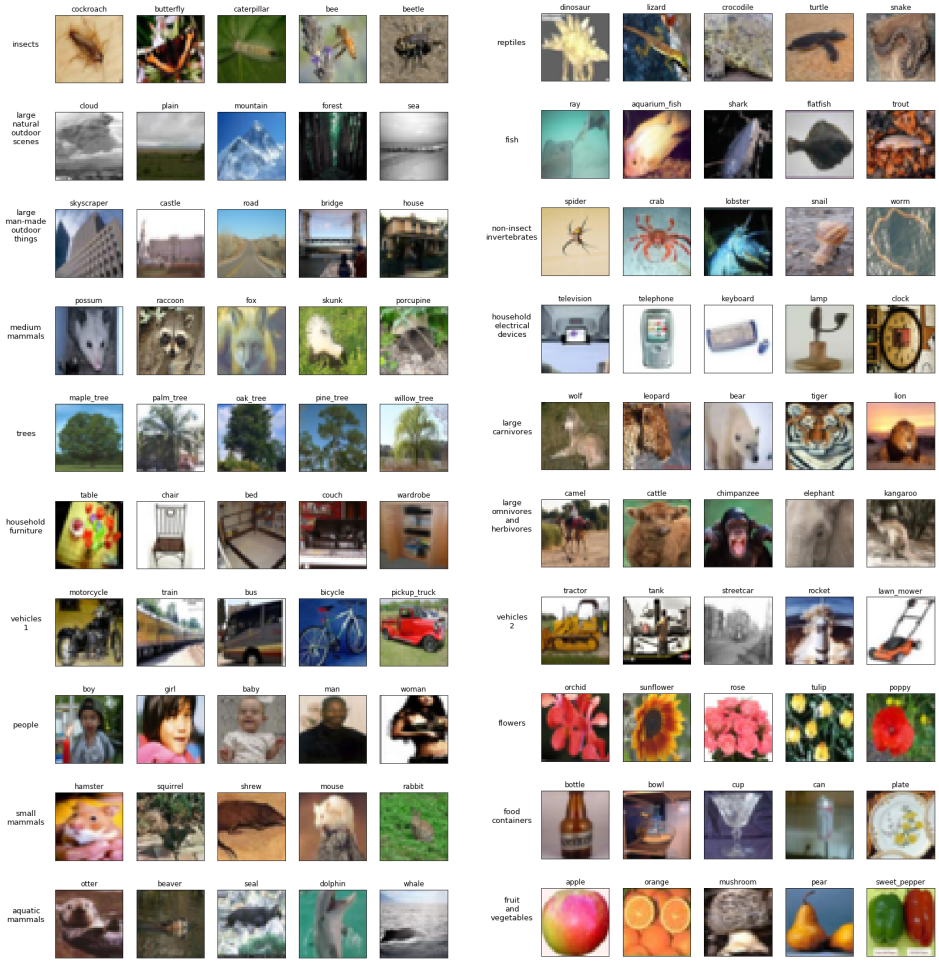

Figure 4.1: Randomly Selected Images for each class in the CIFAR 100

Chapter 5

Evaluation

Created baselines and models are evaluated separately and against each other. For measuring the performance of the created baseline and models, we focus on 3 evaluation area: category prediction, search relevance and efficiency:

5.1 Category Prediction Metrics

Category Prediction metrics are included in the evaluation to understand whether the retrieved visually similar images of a given image query are in the same class or superclass.

Since we don't know what would be the best similar candidate images of a given query image, The below two metrics can be considered as the accuracy on the category level.

5.1.1 Class@10

Class@10 metric counts how many of the retrieved similar images have the same label with the query image. Since K is 10 the metric is out of 10, 10 means all of the retrieved candidates are in the same class, 0 means none of the is the same class with the query image.

$$Class@K = \sum_{k=1}^{K} label(k)$$

label(k) checks if the query image and retrieved result's label are the same or not.

$$label(k) = \begin{cases} 1, & \text{if } k_{label} = query_{label} \\ 0, & \text{otherwise} \end{cases}$$

5.1.2 SuperClass@10

SuperClass@10 is the same as Label@10 but checks for the supercategory level of the retrieved similar images and compares it with the corresponding query image's superclass. If the retrieved image and query image superclass matches it counts as 1 and if it doesn't like 0. SuperClass@10 is the sum of the matches for each 10 similar image candidate with the query image.

$$SuperClass@K = \sum_{k=1}^{K} superlabel(k)$$

super label(k) checks if the query image and retrieved result's super label are the same or not.

$$superlabel(k) = \begin{cases} 1, & \text{if } k_{superlabel} = query_{superlabel} \\ 0, & \text{otherwise} \end{cases}$$

5.2 Search Relevance Metrics

One of the most crucial criteria of any search task is the relevancy of the retrieved search results to the query. Search relevance metrics aim to understand how relevant are the retrieved images and query image.

One of the challenges to understanding the relevancy is that for CIFAR-100 dataset, we didn't have the true results: we don't know which similar candidates should be retrieved for a given query image. That is why we use baseline models as the source of truth to compare the performance of the models created.

The search relevancy metrics are created to compare models with their corresponding baselines.

5.2.1 Intersection@10

To calculate intersection@10 for a given query image, 2 retrieved candidate lists are compared. One of the candidate lists is from a model created and the other one is from its corresponding baseline. Intersection@10 shows the number of matching retrieved similar image candidates between the model and baseline results out of 10. Intersection@10 doesn't take into account the order of the similar candidates in the retrieved lists, it only considers if a retrieved image in a baseline is also retrieved by the model.

Intersection@10 can briefly put as a measurement for results retrieved by model who is also in the baseline.

$$Intersection@K = \sum_{k=1}^{K} rel(k)$$

rel(k) is the relevancy of the kth image in the retrieved image list. If the kth image is in the baseline list then the retrieved result is relevant hence rel(k) is 1, otherwise, it is 0.

$$rel(k) = \begin{cases} 1, & \text{if k is in baseline list} \\ 0, & \text{otherwise} \end{cases}$$

5.2.2 AP@10 and MAP@10

The ranking is another important criteria of the search engines. The visually more similar images need to rank in the first results than the less similar images. AP@10, Average Precision takes into account the ranking of the results while comparing the true list and the retrieved search results list.

Later, MAP@10, Mean Average Precision at 10 for a model is computed to understand the overall performance of a model. MAP is a common metric used in retrieval systems including visual search area [27, 36].

Then, AP@K is calculated as:

$$AP@K = \frac{1}{m} \sum_{k=1}^{K} (P(k) * rel(k))$$

P(k) is the precision at cut off k images. P(k) is the precision calculated from rank 1 to k images in the given list.

MAP@K is calculated as:

$$MAP@K = \frac{\sum_{q=1}^{Q} AP@K(q)}{Q}$$

5.3 Efficiency Metrics

Efficiency is one of the most important metrics when it comes to having a practical and real-life applicable model. For visual search or any other information retrieval tasks, time is a crucial metric to measure. Speed-Accuracy trade-off is unavoidable.

5.3.1 Index and Search Time

Index time is only kept for the model frameworks during the dimensionality reduction step. The time is kept during the transformation process of the given feature vectors in the dimensionality reduction model.

Search time is start being kept just before the creation of the nearest neighbor model until the nearest neighbor returns the closest 10 neighbors for each of the query images in the dataset.

Both of the time metrics are recorded in seconds.

5.4 Letter-Value Plots

Boxen plots, also named as Letter-value plots, are used to show the plot results rather than boxplot because letter-value plots addresses the 2 shortcomings of boxplots: (1) conveys more information about tails by showing quantiles beyond the quartiles, and (2) outliers are labeled and all features shown are actual observations by following Tukey's original boxplot principles [14].

Chapter 6

Experiments & Results

Experiments & Results chapter contains 3 experiments and their results to answer the research questions.

During the experiments, test data is used for querying the images and train data is used as the catalog database. Therefore, the results are obtained with 10000 image queries, 100 queries per class. The visual search results are retrieved from the database from 50000 images, each class having 500 images.

6.1 Experiment 1 - Baselines

6.1.1 Experiment 1 Summary

Experiment Definition:

The first experiment is where are baselines compared to find the best feature extraction types for visual search.

Research questions to answer:

Which image feature representation method is better in the context of visual search according to the defined evaluation metrics?

Created models:

For each feature extraction method, we created one baseline model, and a total of 10 baseline models are created to answer the first research question.

6.1.2 Experiment 1 Results

The results are evaluated based on category prediction and efficiency metrics. Search Relevance cannot be calculated for this experiment since we need a list of best candidate images and it is not available from the dataset.

This experiment is also an attempt to find the best candidate images. The best baseline results will later be assumed and used as the best candidate images, hence the baseline gold standard for the project.

Category Prediction

Out of 10 similar image neighbors retrieved for a given image query, Class@10 checks how many of them have the same class as image query and SuperClass@10 checks how many of them have the same superclass as image query. Therefore, the closer to category prediction metrics, the better is the results for a given image query. Also, the closer distribution of these metrics to 10, the better category prediction results are for the created model.

Figure 6.1 and 6.2 shows the Class@10 and SuperClass@10 Boxen plots respectively. As stated in the Evaluation Chapter, Boxen plots are similar to boxplots and they show the nonparametric representation of distribution and everything shown corresponds to actual observations. It shows also the quantile information where we can understand better the shape of the distribution, especially in tails.

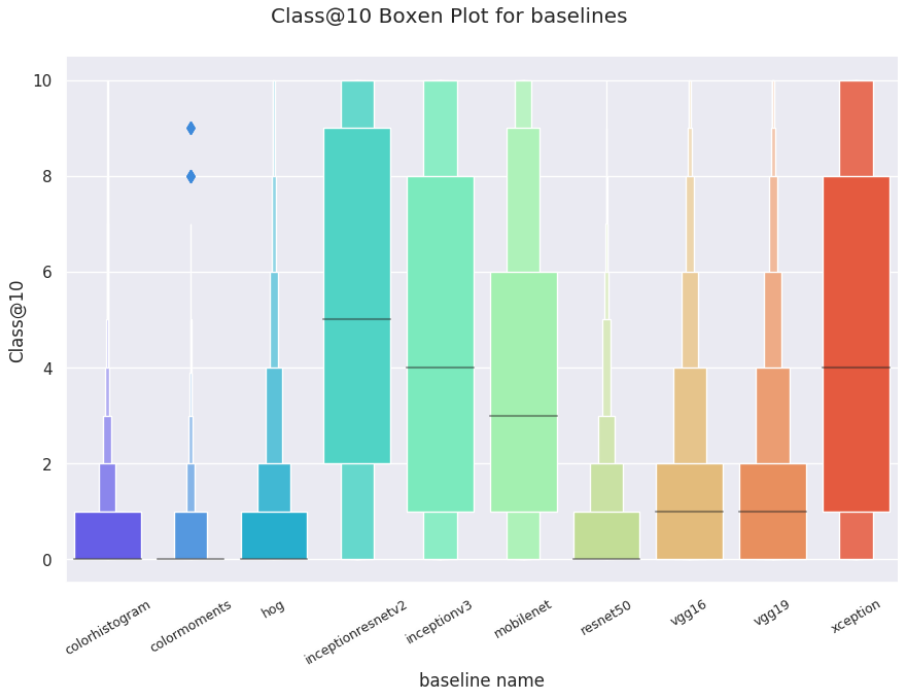

Figure 6.1: Class@10 Boxen Plot for Baselines

Figure 6.1, Class@10 results show that the extracted features from the Inceptionresnetv2 network are the best ones on finding granular class results

matching with the query image with a 50th percentile 5 and a mean 5.2. The worst performing feature extraction method is Color Moments with a 50th percentile 0 and a mean 0.3.

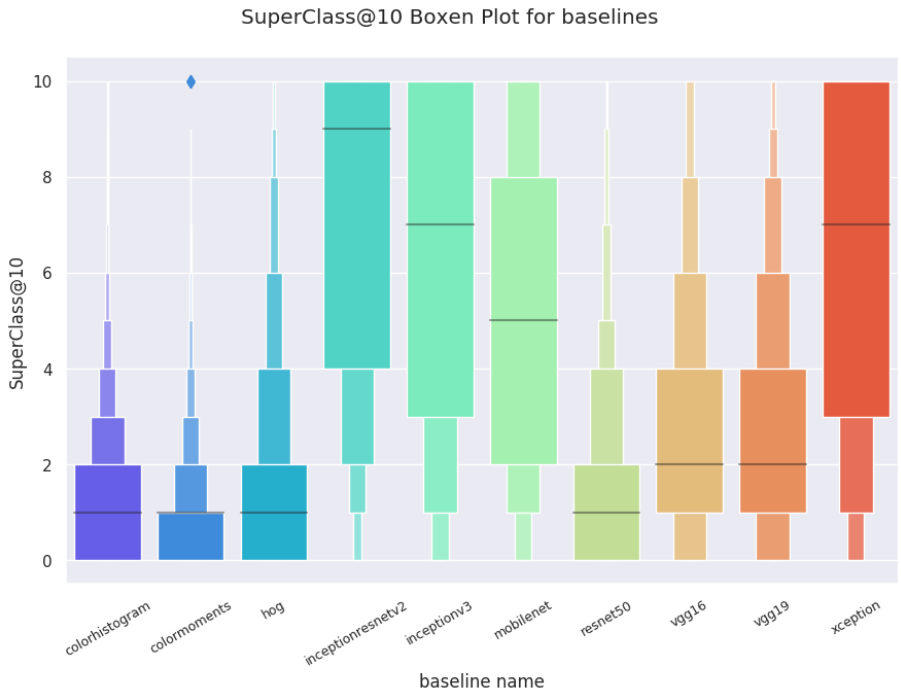

Figure 6.2: SuperClass@10 Boxen Plot for Baselines

Figure 6.2, SuperClass@10 results have the same high level results as Class@10. The order of the performance for baselines are the same. The best-extracted features from Inceptionresnetv2 with 9 on 50th percentile and a mean of 5.2 and the worst is from Color Moments with a mean 0.3. On the other hand, in SuperClass the overall mean and mean per baseline is increased in all of the extracted features compared to Class@10 measures. This shows that even if the retrieved results are not in the same class, the retrieved results are confused with the classes that are in the same superclass. For example, for inceptionresnetv2 results, for 10K image queries, for each query on average 5.3 of the 10 images retrieved are from the same class and 6.9 of these results are in the same superclass with the query image. Table 6.1 and 6.2 shows more information about the distributions of the results.

Baseline Name	mean	std	min	25%	50%	75%	max
colorhistogram	0.56	1.10	0.0	0.0	0.0	1.0	10.0
colormoments	0.30	0.71	0.0	0.0	0.0	0.0	9.0
hog	0.81	1.72	0.0	0.0	0.0	1.0	10.0
inceptionresnetv2	5.30	3.68	0.0	2.0	5.0	9.0	10.0
inceptionv3	4.68	3.60	0.0	1.0	4.0	8.0	10.0
mobilenet	3.66	3.24	0.0	1.0	3.0	6.0	10.0
resnet50	0.83	1.42	0.0	0.0	0.0	1.0	10.0
vgg16	1.57	2.16	0.0	0.0	1.0	2.0	10.0
vgg19	1.56	2.19	0.0	0.0	1.0	2.0	10.0
xception	4.59	3.57	0.0	1.0	4.0	8.0	10.0

Table 6.1: Class@10 Description Table for Experiment 1

Baseline Name	mean	std	min	25%	50%	75%	max
colorhistogram	1.28	1.55	0.0	0.0	1.0	2.0	10.0
colormoments	0.92	1.22	0.0	0.0	1.0	1.0	10.0
hog	1.48	2.11	0.0	0.0	1.0	2.0	10.0
inceptionresnetv2	6.93	3.45	0.0	4.0	9.0	10.0	10.0
inceptionv3	6.39	3.54	0.0	3.0	7.0	10.0	10.0
mobilenet	5.10	3.39	0.0	2.0	5.0	8.0	10.0
resnet50	1.68	1.97	0.0	0.0	1.0	2.0	10.0
vgg16	2.73	2.67	0.0	1.0	2.0	4.0	10.0
vgg19	2.69	2.66	0.0	1.0	2.0	4.0	10.0
xception	6.27	3.54	0.0	3.0	7.0	10.0	10.0

Table 6.2: SuperClass@10 Description Table for Experiment 1

Efficiency

As for the efficiency measure, for this experiment, we only look for search times. Baselines don't use dimensionality reduction methods therefore index time metric is not applicable for baselines. Table 6.3 shows the extracted feature vector dimensions and search times in sec for 10000 image queries per baseline.

Baseline Name	Dimensions	Search Time
colormoments	12	8.75
colorhistogram	300	18.29
vgg16	512	23.71
vgg19	512	24.08
mobilenet	1024	39.39
inceptionresnetv2	1536	53.63
hog	1568	55.10
xception	2048	67.18
inceptionv3	2048	68.52
resnet50	2048	69.85

Table 6.3: Search Time Table for Experiment 1

As we all know, search time depends on the dimension of the features. For brute force KNN, the time complexity is $O(n \times m)$ where n is training samples and m is the number of feature dimensions. Therefore, the higher the dimensions, the more time it takes to retrieve the results, with the current baselines the quickest retrieval takes 9 seconds and the most it takes 70 seconds to query 10K images and find 10 similar images for each image.

Best and Worst Performing Categories

For category prediction metrics, we checked what are the best and worst-performing classes and superclasses per baseline models. In Table 6.4 we share the results for best and worst labels for baseline.

Baseline Name	best_1	best_2	best_3	best_4	best_5
colormoments	orange 1.67	apple 1.27	poppy 1.01	cloud 0.7	plain 0.65
inceptionresnetv2	chair 8.6	keyboard 8.43	clock 8.23	chimpanzee 8.2	orange 8.18
Baseline Name	**worst_1**	**worst_2**	**worst_3**	**worst_4**	**worst_5**
colormoments	television 0.1	otter 0.11	turtle 0.11	flatfish 0.12	mouse 0.12
inceptionresnetv2	shrew 2.21	otter 2.26	girl 2.27	crocodile 2.37	boy 2.63

Table 6.4: Experiment 1 Class@10 Best and Worst labels per Baseline

Baseline Name	best_1	best_2	best_3	best_4	best_5
colormoments	flowers 1.808	fruit and vegetables 1.55	trees 1.304	large man-made outdoor things 1.086	household_ electrical devices 0.98
inceptionresnetv2	household furniture 8.364	food containers 8.178	fruit and vegetables 7.988	household electrical devices : 7.96	trees 7.85
Baseline Name	**worst_1**	**worst_2**	**worst_3**	**worst_4**	**worst_5**
colormoments	food containers 0.616	reptiles 0.666	fish 0.714	insects 0.718	small mammals 0.728
inceptionresnetv2	reptiles 4.908	non-insect 5.312	small mammals 5.678	aquatic mammals 5.706	insects 5.936

Table 6.5: Experiment 1 SuperClass@10 Best and Worst labels per Baseline

In Appendix, section A.1 you can find the results for all baselines.

Query Example and Retrieved Results for each Baseline

In all baselines, the most frequently found label as being worse performed is otter from aquatic mammals superclass and best performed one is the chair from household furniture superclass. Figure 6.3 shows a randomly selected chair image from test data and the retrieved results for each baseline for the image.

As can be seen from Figure 6.3 color features can only capture the similar color tones of the query image but the object classes don't match. Another important takeaway is that not all of the pre-trained networks seem to work well on the visual search, the Class@10 and SuperClass@10 results of Resnet50 and VGG19 once again proven with the image results that they don't perform well compared to the other CNN pre-trained network features. For this specific example, MobileNet and Xception seem to have better qualitative results on retrieved images than InceptionResnetV2.

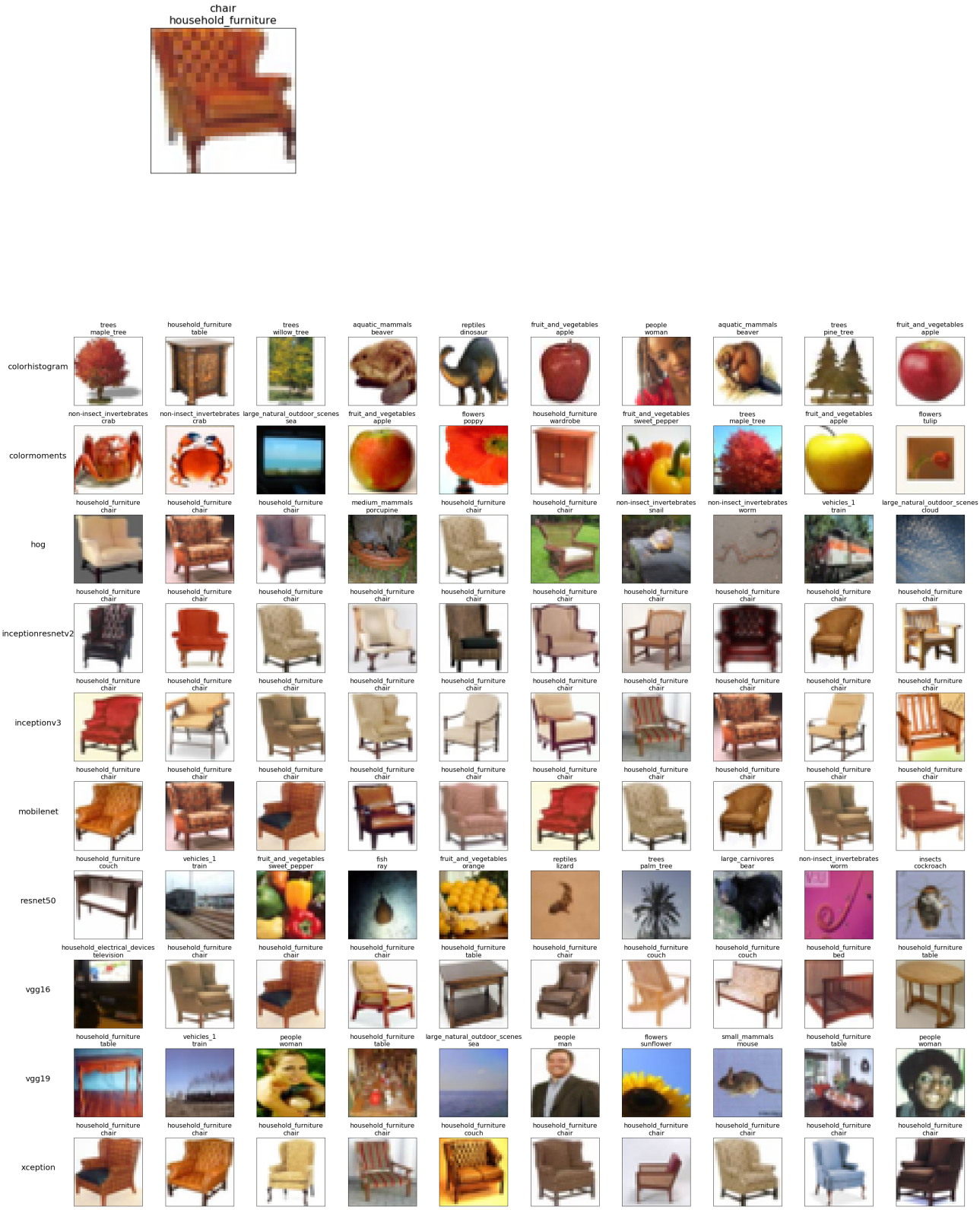

Figure 6.3: Random Chair Image Query and Retrieved Similar Images per Baseline

Same retrieved results is done for a random otter image as well, you can find in Appendix section A.1 Figure A.1. Based on the results shared in the Appendix, we can say that the images where the class is hard to identify hurt from the category prediction metrics since the results retrieved are not from similar classes however the images retrieved seem to capture the characteristics of the given image query.

The result of this experiment, as expected, from worst to best, color features are the worst-performing, followed by HOG and then CNN features. Hence, basic feature extraction methods are performing worse on both class and superclass predictions than more complex feature extraction methods. In the Qualitative results, it is seen that some of the networks might perform worse than HOG.

6.2 Experiment 2 - Baselines: Basic Features vs Combined Features

6.2.1 Experiment 2 Summary

Experiment Definition:

Experiment 2 creates additional baselines with combined features to understand if visual search performance can be improved both quantitatively and qualitatively by combining different feature extraction method results. All of the features are normalized before combining them.

The combined baselines are created with the following 4 ideas in mind:

- Group 1: Combining simple features (color moments, color histogram, hog)
- Group 2: Combining the best CNN feature found in experiment 1 (InceptionResnetv2) with simple features.
- Group 3: Combining the best CNN feature found in experiment 1 (InceptionResnetv2) with other CNN features.
- Group 4: Combining average performing CNN feature found in experiment 1 (MobileNet) with simple features.

Research question to answer:

The main research question to answer in this experiment is: ”Do combining different feature extraction methods improve the results in visual search?”

Group 1 focuses on seeing whether combining less complex features improves visual search performance. Group 2 focuses on understanding if a well-performing CNN feature’s performance can be improved by combining with simple features such as color features. Group 3 seeks an answer

to the question: Do combining different extracted CNN features improve visual search results? Group 4 focuses to answer the same question with 2 except that we change the well-performing CNN feature with an average performing one.

Created models:

In addition to the created 10 baseline models from only one feature extraction method (in experiment 1), additional baseline models are created to answer the second research question of this experiment: can combining models improve visual search results?

For Group 1, 4 models, for group 3, 6 models and groups 2 and 4, 3 models each created. In total 16 new combined baselines are created for second experiment.

6.2.2 Experiment 2 Results

The features used in experiment 1 is abbreviated and concatenated with "-". For example, the ch-hog baseline means the baseline is created by combining Color Histogram and HOG features. The abbreviations can be found in the footnote [1].

Category Prediction

Compared to the experiment 1 baselines, Group 1 and Group 3 combined feature baselines show improvements whereas Group 2 and 4 don't show a significant improvement in the category prediction distribution. Inres2-mn and Inres2-inv3 models, which is the combination of 2 CNN feature extraction methods: InceptionResnetv2, Mobilenet, and InceptionResnetv2, Inceptionv3 networks, seems to improve the Class@10 and SuparClass@10 distributions in the test dataset compared to the InceptionResnetv2 model.

[1] cm: color moments, ch: color histogram, hog: Histogram of Gradients, inres2: InceptionResNetv2, inv3: Inceptionv3, mn: MobileNet, xcp: Xception, res50: ReNet50

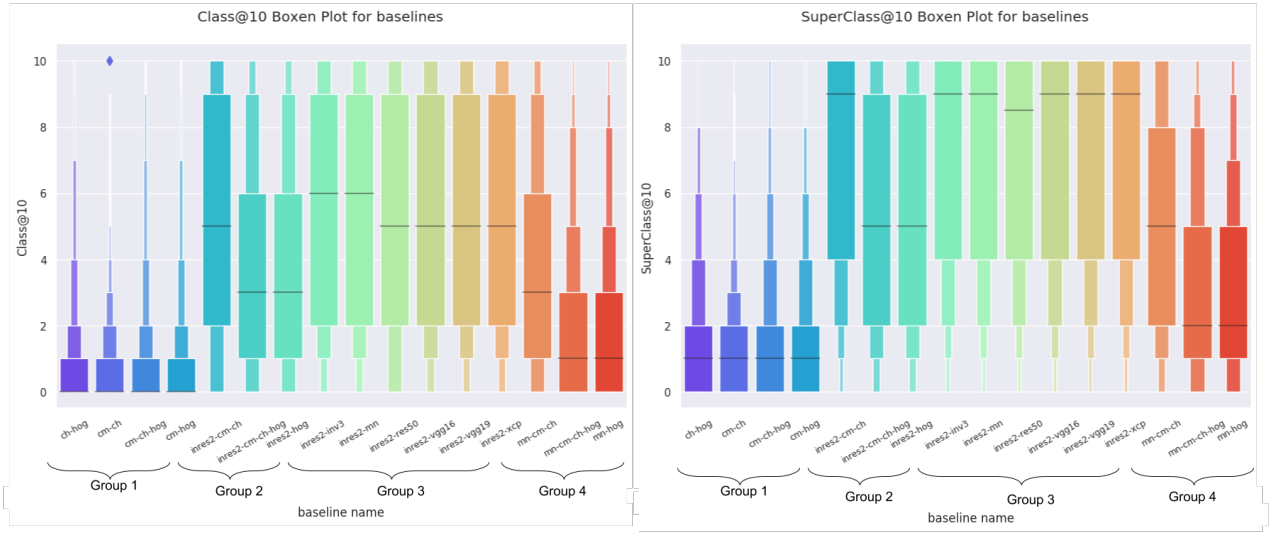

Figure 6.4: Class@10 and SuperClass@10 Boxen Plots for Combined Feature Baselines

From Group 2 and 4 results, we can see that combining CNN features with color or HOG features results in poorer performance in terms of category prediction. In Group 1, combining simple features seems to improve its category prediction ability. In Group 3, we can see that combining different CNN features also seem to improve the results. The first general conclusion could be combining features with their domain, CNN features with CNN, color features with color features seems to give a better category prediction results.

Once investigating the distributions further, we can see some slight improvements with some of the baseline models. Figure 6.6 and 6.7 shows the combined baseline models in more detail. For simplicity reasons, these tables only show the combined baseline models whose distribution's mean is better than all of its basic features' corresponding baseline. Hence the worse or equally performing baseline results aren't included in this table, the whole version of these tables with all of the combined models and their distribution measurements can be found in the tables A.5 and A.6 in Appendix. The rows are highlighted with yellow if the mean is slightly improved (value increased in the hundredths, second decimal point) and highlighted if the mean is significantly improved (value increased in the tenths, first decimal point).

Baseline Name	mean	std	min	25%	50%	75%	max
ch-hog	0.92	1.87	0.0	0.0	0.0	1.0	10.0
cm-ch	0.63	1.20	0.0	0.0	0.0	1.0	10.0
cm-ch-hog	0.93	1.88	0.0	0.0	0.0	1.0	10.0
cm-hog	0.91	1.87	0.0	0.0	0.0	1.0	10.0
inres2-cm-ch	5.32	3.68	0.0	2.0	5.0	9.0	10.0
inres2-inv3	5.45	3.68	0.0	2.0	6.0	9.0	10.0
inres2-mn	5.49	3.59	0.0	2.0	6.0	9.0	10.0
inres2-vgg16	5.35	3.67	0.0	2.0	5.0	9.0	10.0
inres2-vgg19	5.35	3.67	0.0	2.0	5.0	9.0	10.0
inres2-xcp	5.43	3.65	0.0	2.0	5.0	9.0	10.0

Table 6.6: Class@10 Description Table for Experiment 2

Baseline Name	mean	std	min	25%	50%	75%	max
ch-hog	1.63	2.24	0.0	0.0	1.0	2.0	10.0
cm-ch	1.39	1.70	0.0	0.0	1.0	2.0	10.0
cm-ch-hog	1.64	2.25	0.0	0.0	1.0	2.0	10.0
cm-hog	1.62	2.24	0.0	0.0	1.0	2.0	10.0
inres2-inv3	7.08	3.41	0.0	4.0	9.0	10.0	10.0
inres2-mn	7.06	3.33	0.0	4.0	9.0	10.0	10.0
inres2-vgg16	6.95	3.44	0.0	4.0	9.0	10.0	10.0
inres2-vgg19	6.95	3.44	0.0	4.0	9.0	10.0	10.0
inres2-xcp	7.02	3.41	0.0	4.0	9.0	10.0	10.0

Table 6.7: SuperClass@10 Description Table for Experiment 2

Although it is a necessary measure, it is important to keep in mind that category prediction metrics are more related to the classification of the object in the images rather than the semantics of images. For example, none of the category prediction metrics can capture the similarity of images of having similar classes with different texture and patterns. Therefore, not seeing significant improvement on these metrics doesn't necessarily mean search relevance doesn't improve. Or having significant improvement can mean that we bias the model to object classification and it might risk having results with more visual similarities. We can further investigate the change by looking for some random query images and the retrieved results by different combined models to see some qualitative results.

Efficiency

The drawback of combining features is that it increases the search time since KNN depends on the dimension of the data.

Figure 6.8 shows the new dimensions of the combined features and the time it took to query 10K images and retrieving 10 similar image results for each of them. The search results of Experiment 2 are unsurprisingly increased and ranges from 19 seconds to 2 minutes. In Experiment 1, the Inception-ResnetV2 dimension was 1536 and it took 53.63 seconds to perform 10K search whereas with a better performing Inres2-mn combined baseline this time is increased to 81.17 seconds.

Baseline Name	Dimensions	Search Time
cm-ch	312	18.21
mn-cm-ch	1336	46.49
cm-hog	1580	52.71
inres2-cm-ch	1848	60.04
ch-hog	1868	61.17
cm-ch-hog	1880	61.16
inres2-vgg16	2048	66.23
inres2-vgg19	2048	65.59
inres2-mn	2560	81.17
mn-hog	2592	83.17
mn-cm-ch-hog	2904	91.11
inres2-hog	3104	95.16
inres2-cm-ch-hog	3416	104.37
inres2-xcp	3584	108.31
inres2-inv3	3584	108.78
inres2-res50	3584	118.06

Table 6.8: Search Time Table for Experiment 2

Another question to answer is then if combining features is a good trade-off because it might increase the category prediction and potentially the search relevance but it will cost to wait for more for a better result. This is an important aspect to keep in mind always since search time increases linearly with the dimension of the data and in the end, having the best search relevance might cost the model to be not practical to be used in real life.

Example Queries

Figures 6.5, 6.6, and A.9 show some query examples and their retrieved results in the InceptionResnetV2 baseline and combined baselines that the category prediction results improved.

The examples are chosen randomly, and more can be found in Appendix Experiment2 section. The first row shows the simple baseline model: InceptionResnetv2 results (found to be best in Experiment 1). The remaining rows are some of the results found in Group 2 and 4 where combining InceptionResnetv2 with other CNN and color features.

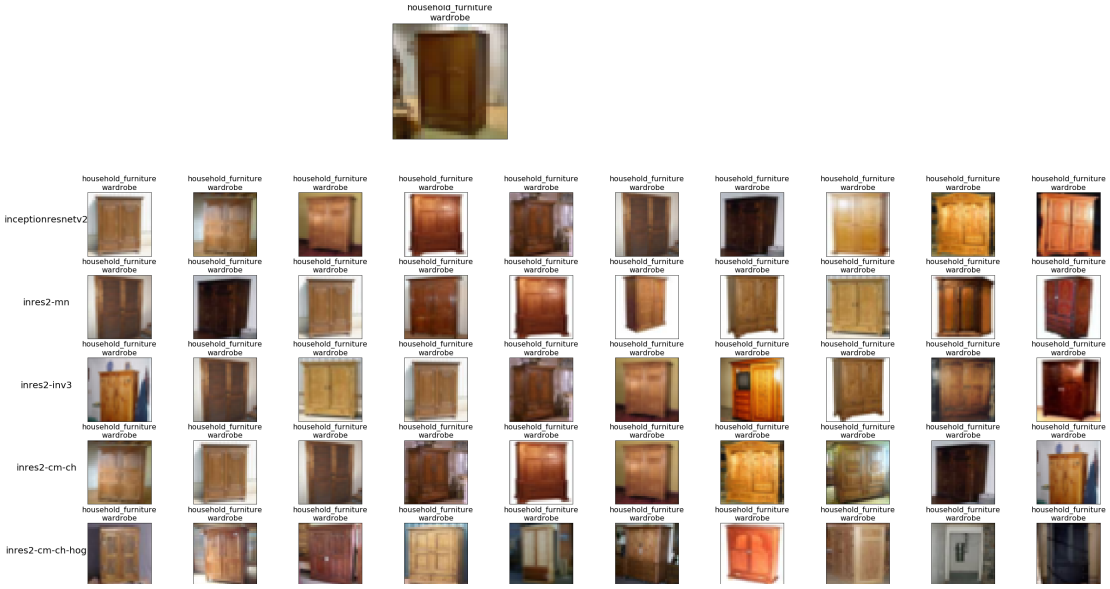

Figure 6.5: Retrieved Results for Wardrobe Query Image

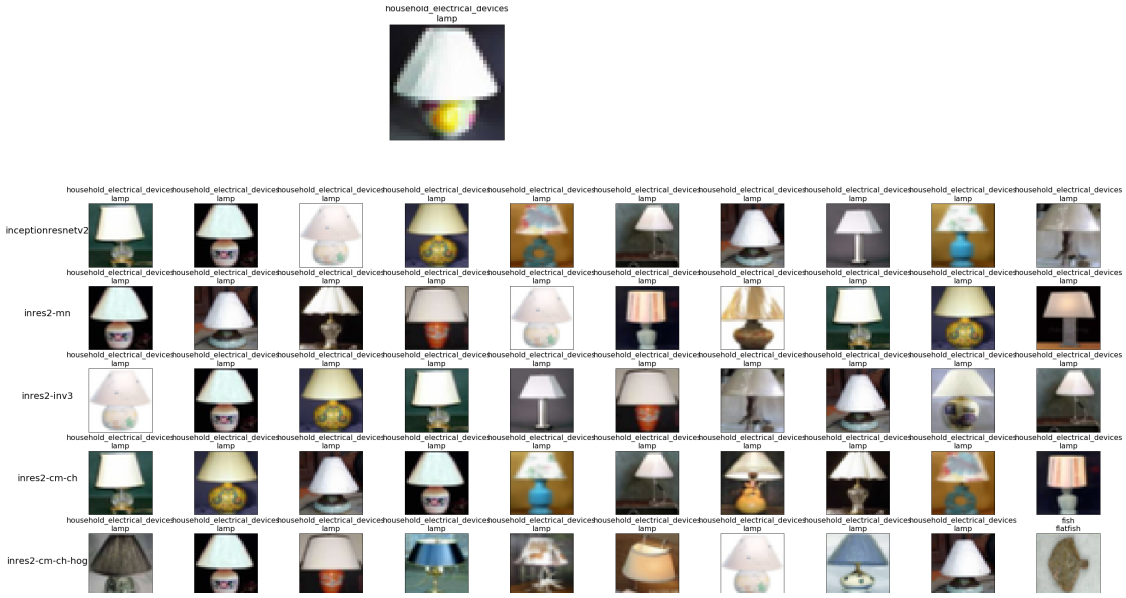

Figure 6.6: Retrieved Results for Lamp Query Image

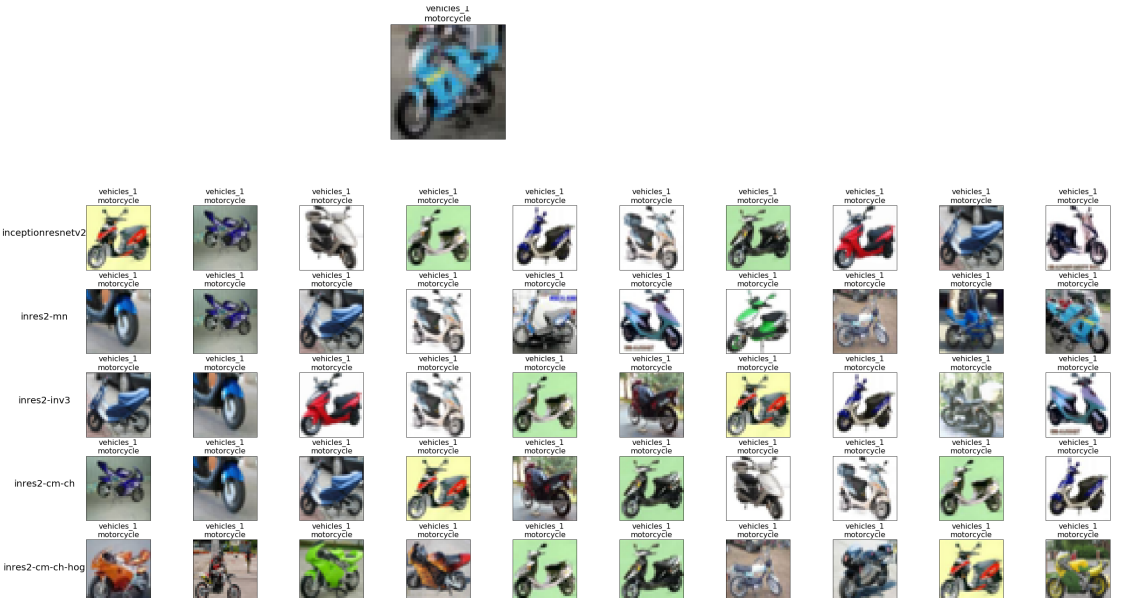

Figure 6.7: Retrieved Results for Lamp Query Image

From the example images, it can bee is seen that combining different fea-

tures, changed the visual search result rankings in all of the images. For wardrobe image, for example, the wood type and color in the combined features seem to rank better for the combined feature inres2-mn than the InceptionResnetV2 baseline the given query results. Inres2-mn seems to have darker wardrobes in the first orders and seems to have more dark-colored wardrobes in the search result. For the Lamp query, again combining features help retrieve more visually relevant results ranked higher for example inres2-inv3 first 3 results and inres2-mn baseline overall returned seem to caught better the sphere-shaped body part of the lamp and white shade. For the motorcycle query, we can see that inres2-mn combined features return 9 out 10 returned motorcycles in blue color as well which seems to be more relevant results overall for the given query image.

To conclude the Experiment 2 results, it is seen that combining different features seem to improve the category prediction for a given baseline but the combination of the features must be selected carefully since some feature combinations might hurt the performance by confusing the model more than helping. The selection can be done both by checking the metric results and also going through the visual search results and comparing it with the initial baselines. 3 of the combined baselines out-perform the InceptionResnetv2 which are Inres2-Inv3, Inres2-Mn and Inres2-Xcp in terms of category prediction.

6.3 Experiment 3 - Baseline vs Models

6.3.1 Experiment 3 Summary

Experiment Definition:

Experiment 3 is about applying dimensionality reduction and creating different versions of the baseline and compare the results with the baseline to see if the dimension reduction can help to find still relevant results with a better time result. Hence, this experiment is more on understanding the accuracy/relevancy vs time trade-off in visual search.

When comparing the baseline with models search relevancy and efficiency metrics will be used. Hence the baseline model will be taken as the expected result and the model's performances will be compared against how close the model performance to the baseline.

There is two retrieval possibility using query images within the catalog (in-shop retrieval) or outside of catalog (consumer to shop retrieval). In this experiment, we used the test data as catalog and query data, since using train data as catalog had exceeded the hardware resources we had and memory capacities for some of the dimensionality reduction method. Since this is a retrieval system, using test images as a catalog can be considered as within-

catalog image retrieval (in-shop retrieval). In the previous experiments, this was made by using out of catalog images where the train image was catalog and test images were query (consumer to shop retrieval). These are two different types of retrieval methods used in industry so our end-to-end system is flexible to handle both types of retrieval. Although, to make sure that they perform the same task and make baseline and models comparable, baseline model recreated to perform in-shop retrieval.

Research question to answer:

The third Experiment's research question to answer is "Should a dimensionality reduction method be preferred over another in the context of visual search?". The idea of applying dimensionality reduction is to try to have more compact feature vectors and hence reduce the search time. This might be especially useful for combined feature baselines since we hurt the search time by increasing the dimensions. This experiment will only focus on to see if applying dimensionality reduction to InceptionResnetV2 baseline feature can still give good and relevant visual search results. Also, if some dimensionality methods perform better than others in the CIFAR 100 dataset within the visual search context.

Created Models:

In this experiment, the InceptionResnetV2 baseline and 5 additional models will be used. These 5 models are each created with different dimensionality reduction methods applied on top of InceptionResnetV2 features. The Dimensionality Reduction methods are PCA, UMAP, Random Projection, LDA and NCA.

To make the models compared to each other each model will be reduced to the same dimension. The dimension is selected by using Johnson-Lindenstrauss Lemma and using its minimum dimension bounds. The InceptionResnetV2 model's dimension is 1536 and the dataset has 60K sample images. According to Johnson-Lindenstrauss Lemma conserving pairwise distances between the data depends on the number of samples and not dimensions. Hence while calculating the minimum number of components necessary we need to use 60K, however since we will use the found number as the new dimensions for dimensionality reduction the number should be lower than 1536 which is the initial number of dimensions. Epsilon values of 0.35, 0.5 and 0.65 are chosen as the distortion rate which gives 937, 528 and 367 (respectively) as the new dimension to reduce. Therefore, for each model the 1536 dimension of the CNN feature is reduced to 937,528 and 367 dimensions, meaning we almost reduced the dimension with a rate of 1.5, 3 and 4.

6.3.2 Projections

To understand better the feature space of InceptionResnetV2, different visualizations of the feature vectors is shown in the 2D plot in Figure 6.8 shared below:

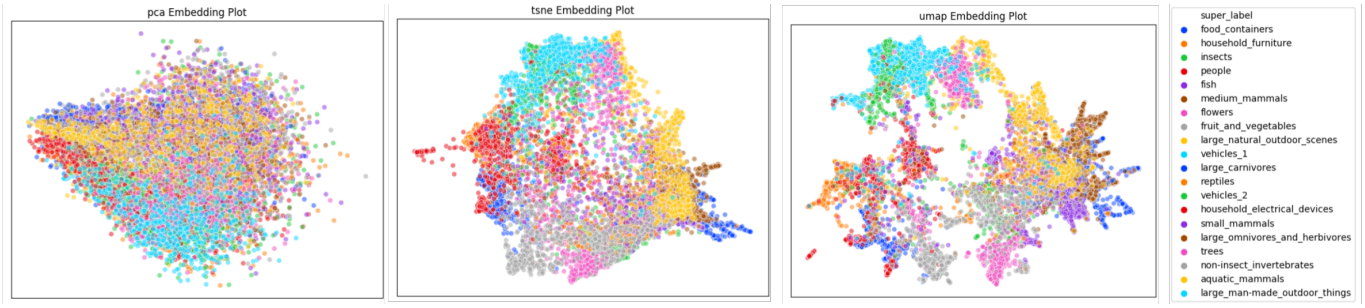

Figure 6.8: PCA, t-SNE and UMAP visualizations of InceptionResnetV2 features

Each plot is colored based on the superclasses of the images to understand whether superclass objects are mapped close to each other in the feature space visualizations. In the PCA plot, according to colors the superclasses are scattered around the plot whereas in the t-SNE plot superclasses are better grouped. In the UMAP plot, we can see that the separation between superclasses is better than the former 2 dimensionality reduction method visualizations. According to the scatter plots, UMAP seems to have a better separation in superclasses for the images. However, it is important to keep in mind that this is only a visualization which the dimensionality reduction is applied to reduce the features into 2 dimensions, hence it might not be representative for the high-dimensional space of InceptionResnetV2 features and nor for the model features where these dimensionality reduction methods are applied. It still gives us a valuable understanding of the data and how the features are mapped together in 2 dimensions.

To understand the feature space better and also how these apply to the class level, we also plot the visualization with labels. The colors indicate the same superclasses (same color code as Figure 6.8) and labels show the class names of the images. Since the dataset is very big, to have a more readable visualization the data is sampled randomly into 5K images to show the Figures 6.9, 6.10 and 6.11. From 6.9, it can be seen that the classes are not grouped well together either. As an intuition, the mapping into a lower-dimensional space might not conserve the distances between pair of images and it might be a high probability that PCA might perform badly in the visual search.

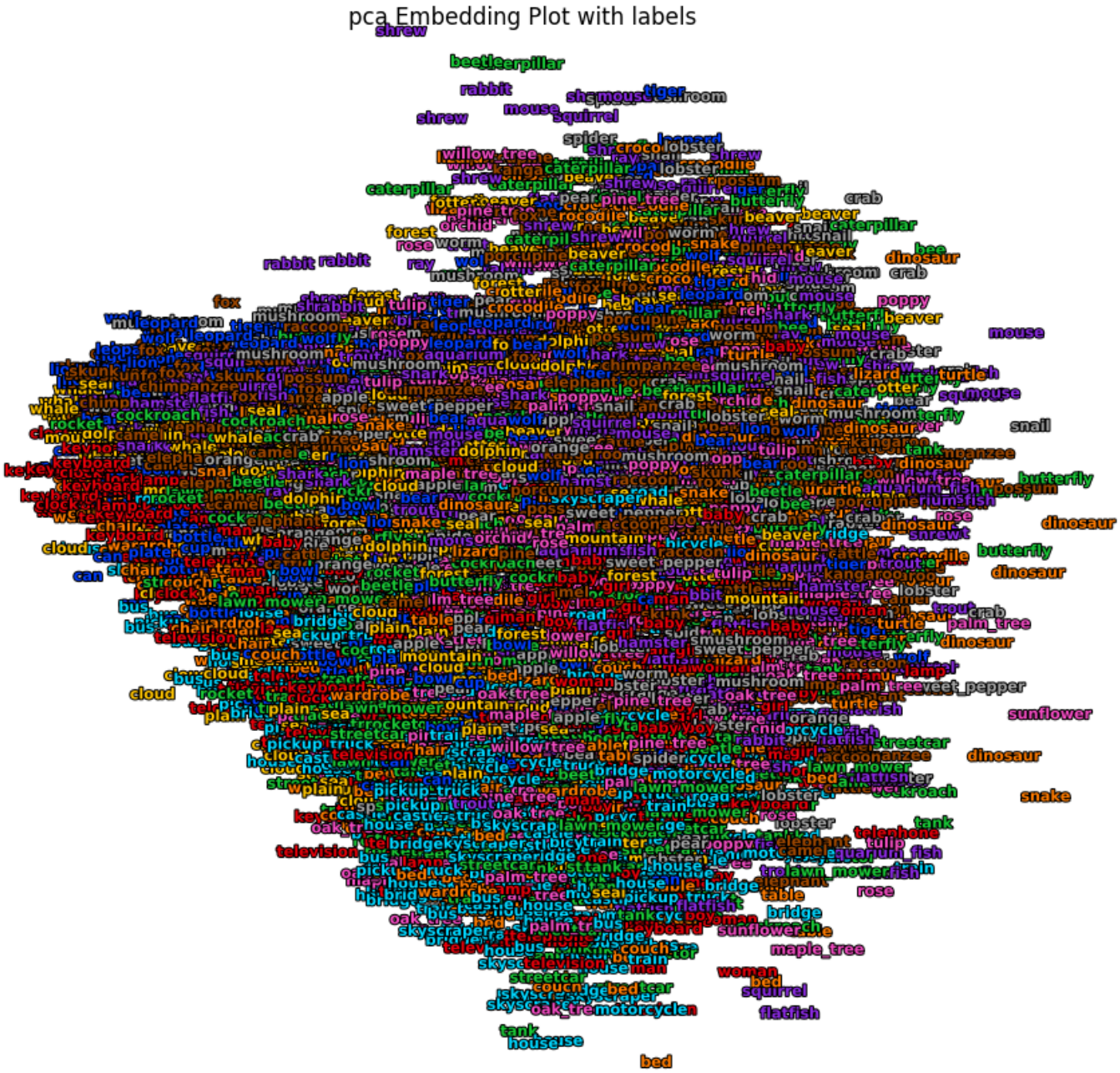

Figure 6.9: PCA Embedding Visualization with Class Labels

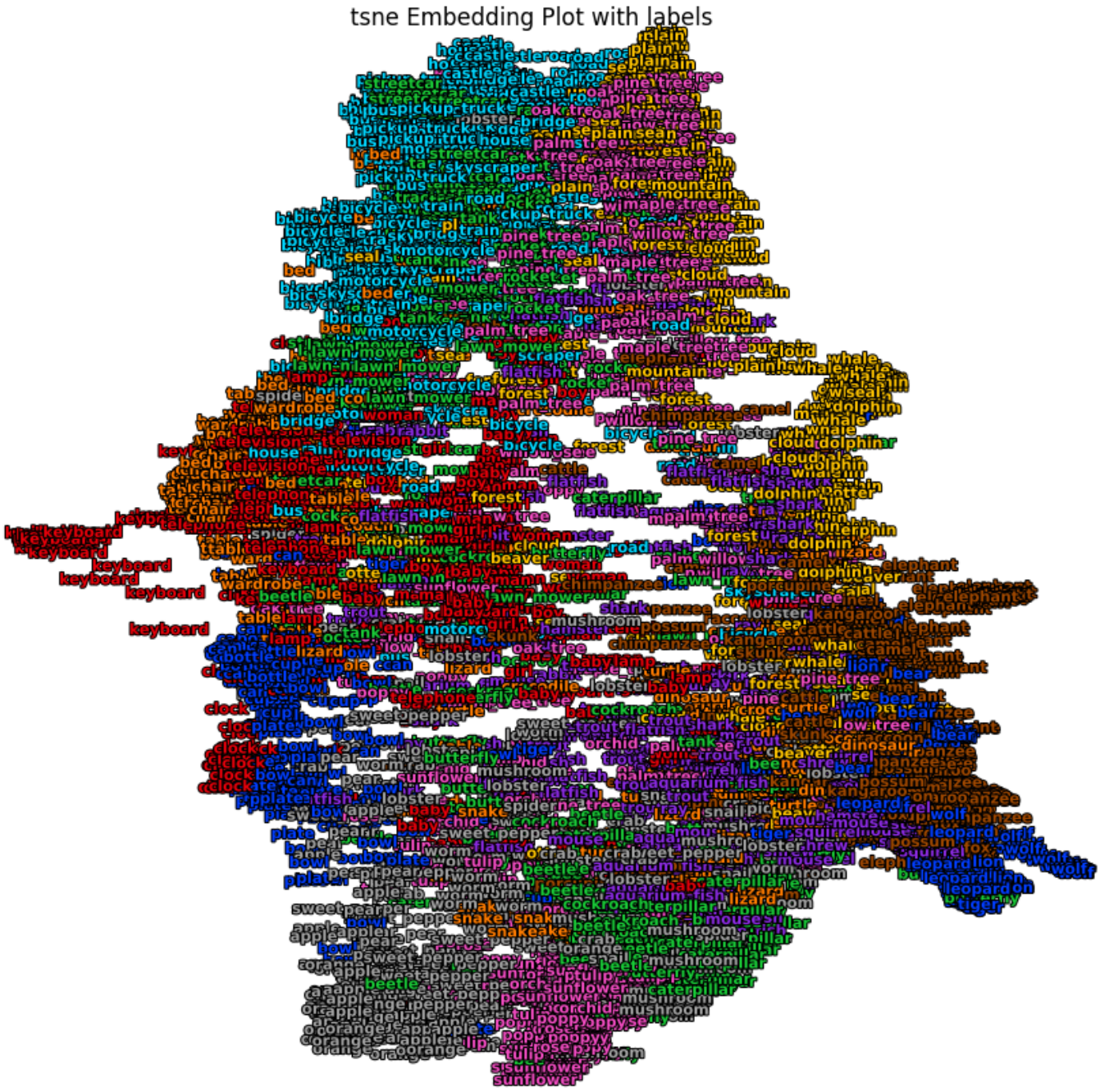

Figure 6.10: t-SNE Embedding Visualization with Class Labels

Figure 6.10 on the other hand seems to have the superclasses grouped well together. Some classes such as keyboard clock are well grouped with the images with their classes however most of the classes are rather spread around

the plot.

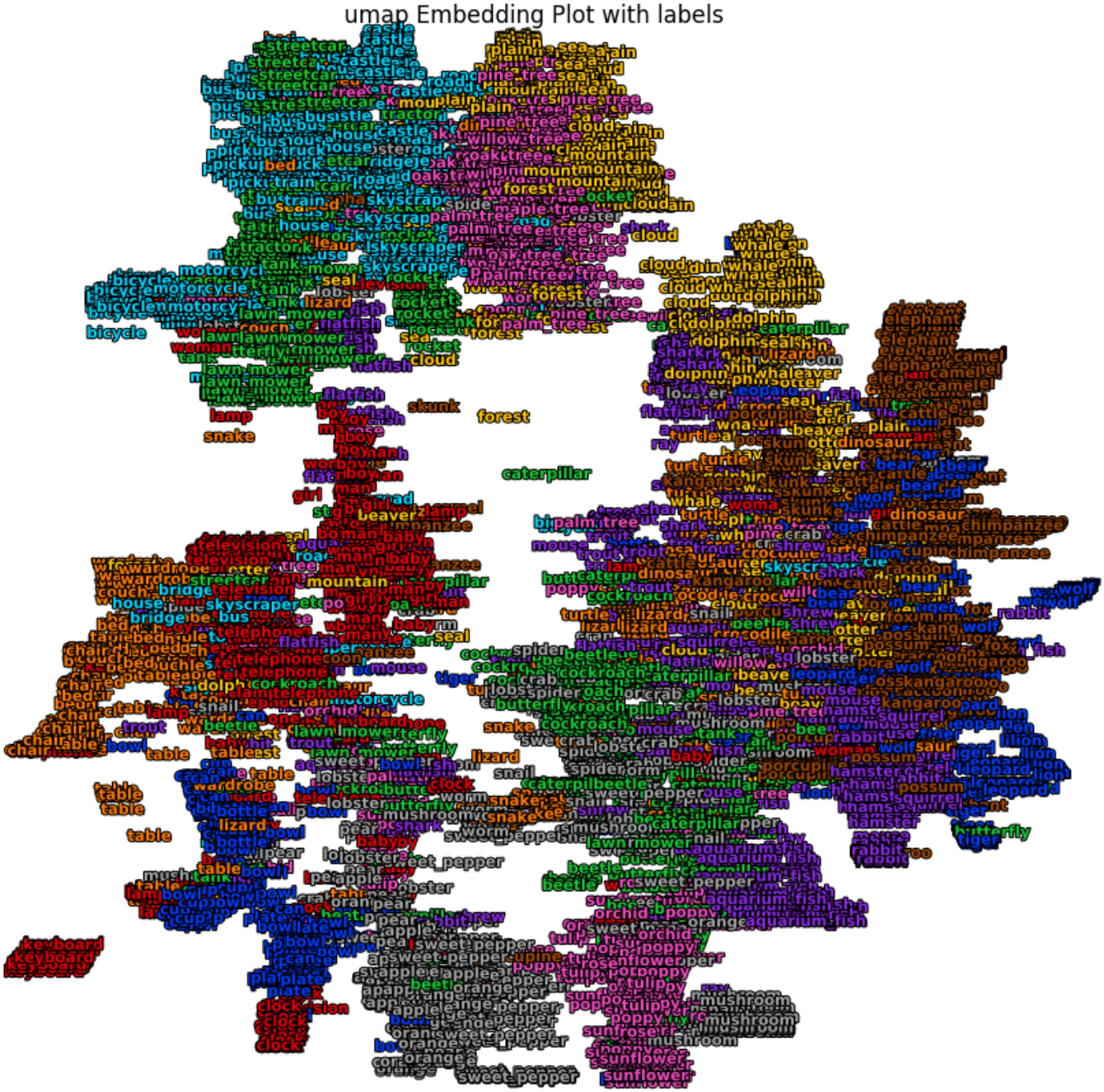

Figure 6.11: UMAP Embedding Visualization with Class Labels

In the last plot, we can see that both superclass and class distributions are much better and classes are grouped way better together than the previous plots. It is easier to spot which classes are mapped close to each other in 2-dimensional UMAP results of InceptionResnetV2 features. For example, in the yellow right side of the plot, most of the animal classes are grouped according to their superclasses whereas on the upper side on outdoor things such as natural ones (plain, mountain, sea) and human-made ones (castle, road, skyscraper) and vehicles are mapped close to each other. On the left side objects and household furniture such as a table, keyboard, and wardrobe are grouped and on the lower side of the plot, the fruit and vegetables and flowers are mapped close together. UMAP did a very good job of separating classes and superclasses well and seems to be promising in having effective dimensionality reduction results for visual search.

One thing to stress is that even though the well class and superclass separation, some images are mapped in a space far away from its class and superclass group. This might be due to image quality, where some of the images are very bad quality and hard to distinguish the class. Another reason is that when applying dimensionality reduction some of the information is lost and this hurts the results by having a lower dimension embedding for certain features where the new distance is closer to other class objects than its own. Or this might be related to the feature extraction process where the closest neighbors of these images were never been from its class. Independently from the reason, this shows that for some images their nearest neighbors in the feature space are not relevant from the start and this might be the case in the original space as well. Meaning, even the gold-standard best results might not be visually very relevant to these images. This is one of the problems that makes visual search task very challenging to solve.

6.3.3 Experiment 3 Results

After having a better intuition on some of dimensionality reduction methods and their visualization, the results for the baseline and models are shared in the next sections:

Search Relevance

For Search Relevance, we take the InceptionResnetV2 model as a baseline and treat it as the best possible result. Then each of the models created with dimensionality reduction compared with the baseline. Search Relevance metrics Intersection@10 checks how many of the images found are also in the baseline's results regardless of their order. MAP@10 does the same but takes the order of the results into account as well. Figure 6.12 shows the relevance results of each model for all of the different dimension results:

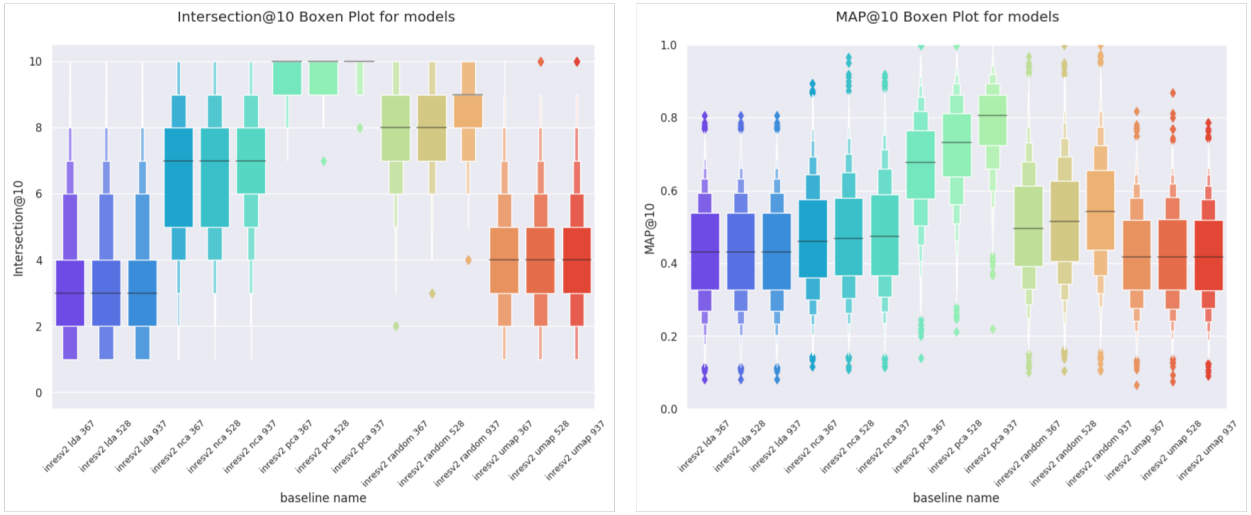

Figure 6.12: Intersection@10 and MAP@10 Boxen Plots

As can be seen the Intersect@10 and MAP@10 results are showing similar best and worst results irrelevant of number dimensions. PCA being best followed by Random Projections, and LDA being the worst one. Therefore for the sake of simplicity, we will show only the model results of dimension type which is the models reduced to 937 dimensions.

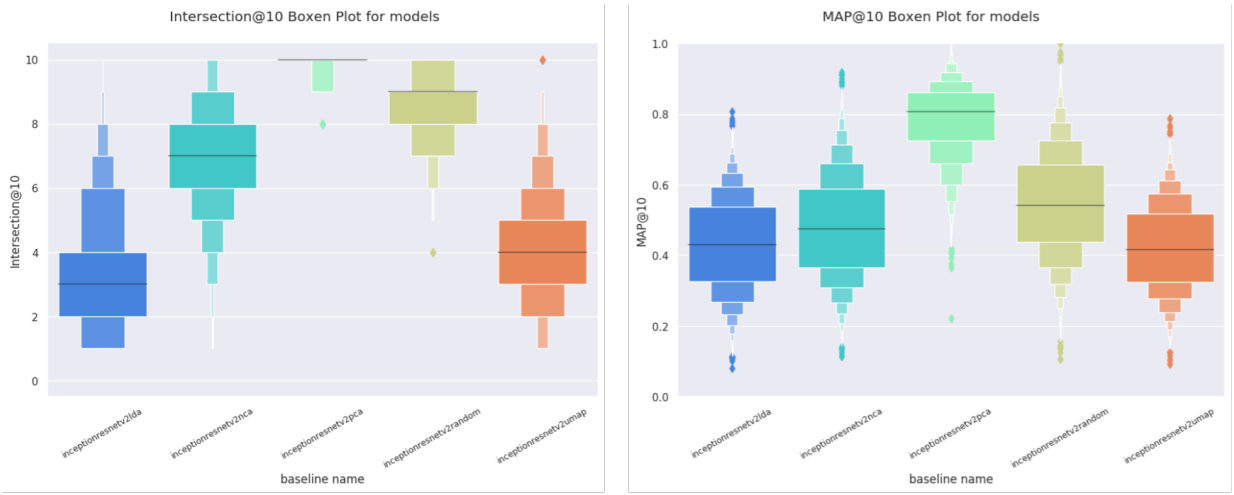

Figure 6.13: Boxen Plots for Intersection@10 and MAP@10 for Models reduced to 937 dimensions

In Figure 6.13, Intersetcion@10 results show that the overall PCA model has

the most matching results with baseline after the dimensionality reduction, followed by Random Projection model. For PCA, almost all of the query images' baseline results are retrieved with PCA, with a mean of 9.88 (see Fig. 6.9). This number is worst with LDA where the mean is 3.22.

Baseline Name	mean	std	min	25%	50%	75%	max
inceptionresnetv2_lda_937	3.22	2.01	1.0	2.0	3.0	4.0	10.0
inceptionresnetv2_nca_937	6.84	1.82	1.0	6.0	7.0	8.0	10.0
inceptionresnetv2_pca_937	9.88	0.33	8.0	10.0	10.0	10.0	10.0
inceptionresnetv2_random_937	8.55	1.05	4.0	8.0	9.0	9.0	10.0
inceptionresnetv2_umap_937	3.99	1.74	1.0	3.0	4.0	5.0	10.0

Table 6.9: Intersection@10 Description Table for Experiment 3

For Visual search, not only the retrieval but the ranking of the results are very important as well. When we look at the MAP@10 results, it can be seen that the PCA model is not only performing well on retrieving similar results to baseline but it also conserves the ranking order of the results. For MAP@10 range is between 0 and 1, 1 meaning all of the results are retrieved and in the same order of the given list. We can see that on average for PCA 8 out of 10 results are retrieved in the same order as it was in the baseline model.

Baseline Name	mean	std	min	25%	50%	75%	max
inceptionresnetv2_lda_937	0.43	0.13	0.080556	0.327083	0.430556	0.538368	0.806250
inceptionresnetv2_nca_937	0.48	0.15	0.114583	0.366667	0.475000	0.588889	0.918750
inceptionresnetv2_pca_937	0.79	0.10	0.220833	0.725000	0.806250	0.862500	1.000000
inceptionresnetv2_random_937	0.55	0.15	0.105556	0.437500	0.541667	0.656250	1.000000
inceptionresnetv2_umap_937	0.42	0.12	0.091667	0.325000	0.416667	0.518750	0.787500

Table 6.10: MAP@10 Description Table for Experiment 3

PCA is an unsupervised linear dimensionality reduction method and performed better than other methods and even from supervised methods. One reason behind this is that relevancy looks for how well a result performed compared to the provided baseline. Our baselines might not be very good and representative enough in the first place. Another reason could also be that supervised methods bias the dimensionality reduction with the classes whereas for some cases as shown above the visually similar images found in the baseline might not have the same classes. However, making a decision

only from quantitative results might be deceiving so we will explore the results in the next sections to check some example images. The results table for all of the dimensions for MAP@10 and Intersection@10 can be found in Appendix Section A.3.

Efficiency

One of the advantages of using dimensionality reduction is to improve the search time. This costs to have an additional index time and decrease in relevancy. However, relevancy can be used as a trade-off and fine-tuned in terms of time efficiency we want and the search relevance performs we want to form a model. Index time introduces extra time at the very beginning once to create the index for the whole catalog but once they are created, the only index time needed is per query which can be more affordable than having a higher search time per query. Because for a given query image index time is calculated once but in search pairwise distances checked one by one for each sample, the lower the number of the dimensions of sample times more efficient the search retrieval time would be.

Table 6.11, shows the index and search time for 10K image queries.

Baseline Name	Dimensions	Index Time	Search Time
inceptionresnetv2_lda_937	99	0.21	2.56
inceptionresnetv2_nca_937	937	0.94	4.77
inceptionresnetv2_pca_937	937	0.52	4.70
inceptionresnetv2_random_937	937	0.49	4.93
inceptionresnetv2_umap_937	937	0.12	3.93
inceptionresnetv2_pca (baseline)	1536	-	6.9

Table 6.11: Experiment 3 Index and Search Times

LDA by its nature reduces the dimensions into the number of classes -1 dimension, that is why we have 99 dimensions for the LDA case. When we checked the efficiency results, we see that the search time results decreased as expected.

Experiment 3 Examples

It is also important to understand the qualitative results and how reducing the dimensions changed the results by investigating some of the examples. The first row is the baseline and the remaining rows are the models created with dimensionality reduction methods and the final dimensions are 937.

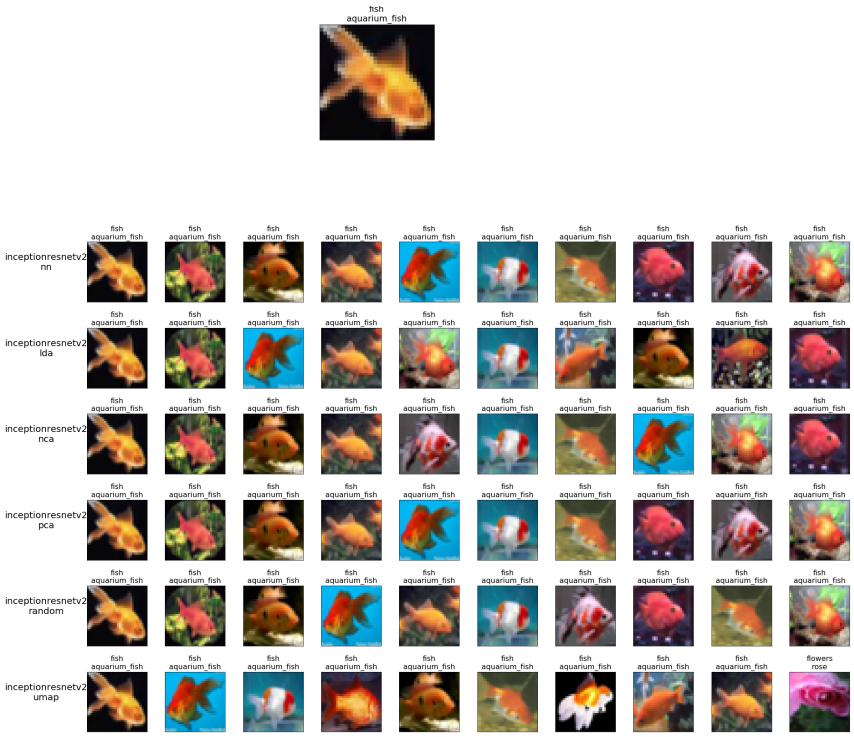

Figure 6.14: Retrieved Results for Aquarium Fish Query Image

We can see that for this image PCA, exactly retrieves the same 10 similar images with the same order. For the rest of the models, the retrieved results seem to be relevant as well except the last image of the UMAP model which retrieves a rose image.

Figure 6.15: Retrieved Results for Baby Query Image

The baby image example is one of the images that illustrate the different behavior of supervised and unsupervised dimensionality reduction methods better. LDA and NCA (the supervised methods) retrieved more baby results than the baseline and the other dimensionality reduction methods. This can clearly show that supervised methods are biased towards the class information while applying dimensionality reduction which could be both an advantage and disadvantage. In this image specifically, LDA results seem to be more relevant hence better than the Baseline results. The disadvantage biasing towards the class information could hurt the results in a way that images visually very similar but without having the same object might go very end of the result orders or not even be retrieved.

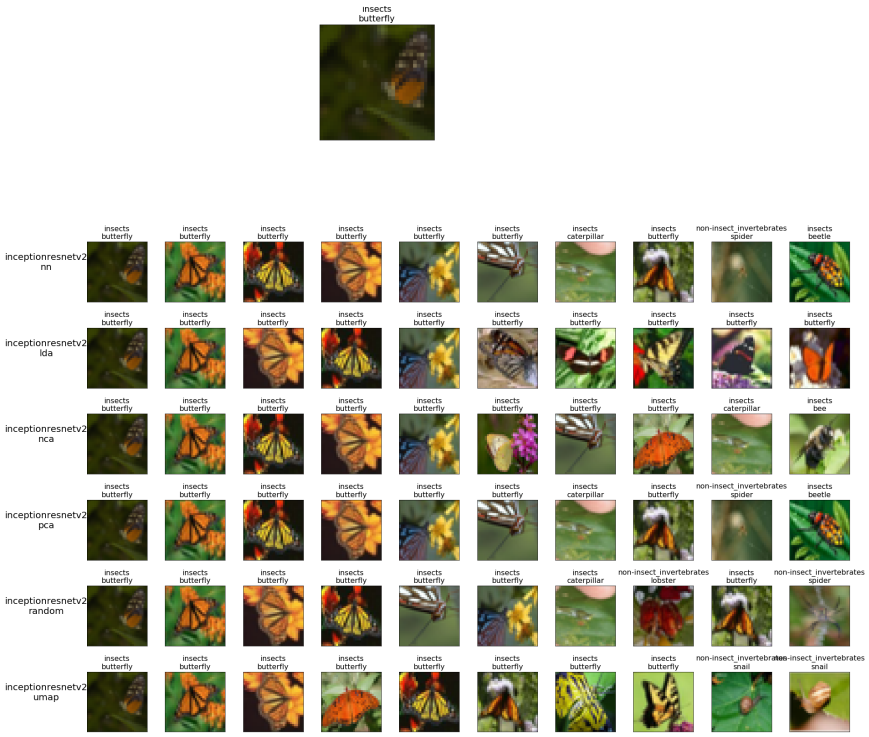

Figure 6.16: Retrieved Results for Butterfly Query Image

Butterfly example is also another similar example to see the distinction between unsupervised and supervised dimensionality reduction methods. LDA and NCA retrieves more In this example, Random Projection and PCA result visually seem to retrieve the worst results even though they retrieved the closest results to the baseline. This also shows that having good baselines is important to measure relevance. Our baseline was one of the well-performing in the previous experiments but it doesn't mean that its results are the best and optimal ones. This is why having a dataset where gold standard results exist to learn visual similarity is important. Then, understanding the relevancy and comparing the models would be more accurate.

More examples can be found in the Appendix, section A.3.

Experiment 3 Conclusions

To conclude Experiment 3, it is seen that the performance of the different dimensionality reduction methods doesn't depend on the dimension numbers. The meaning, one-dimensionality method doesn't perform better than the

other when the dimensions are very low but perform worse when they are high or vice versa. For CIFAR 100 dataset and InceptionResnetV2 features, PCA seems to be the dimensionality reduction technique that performs closest to the created baseline followed by Random Projections. Normally, supervised methods expected to perform better than Random Projections, however, after checking some example images we saw that biasing towards class might hurt the retrieval results if the expected baseline results include results from different classes. In overall results, UMAP seems to have the worst retrieval results and also needs big memory for the calculation which makes it undesirable for this use case. This was also unexpected since it's 2D visualization results look very promising. From the experiments, we conclude that UMAP is performing better for visualizations and might not be suitable for this task or this specific dataset.

One of the important take aways is that baselines are really important and having a good dataset with clear, specified visually similar images are crucial to understanding the created models and baselines and have reliable relevancy results.

Chapter 7

Conclusion & Future Works

To conclude, the main goals of this thesis are: (1) creating a visual search system that is easily modifiable for continuous development, research, and integration; (2) conduct some experiments to understand how different feature extraction methods and dimensionality reduction methods perform for visual search models.

To understand the effects of feature extraction and dimensionality reduction methods 3 Experiments done. The first experiment is to understand if any feature extraction method is better than the others and feature vectors extracted from pre-trained CNN networks performed better than simple color features or handcrafted methods such as Histogram of Gradients. The second experiment focuses on understanding if these extracted features are complementary to each other or alternatives to each other in the context of visual search. 4 different experiment group is formed and different extracted features are combined for the second experiment. As a result, it is seen that combining simple features and CNN features together improved the category prediction results of the visual search models whereas combining features in cross areas such as colors + CNN features either hurt or didn't changed the performance very much. In the last experiment, dimensionality reduction methods are compared against each other to understand if some techniques are better preserving the search relevancy results. PCA and Random Projection methods were the top-performing dimensionality reduction methods that preserve the closest baseline results.

As future work, this thesis can be extended and improved in several directions:

- Including different Feature extraction methods such as SIFT, FAST, ORB, and PHOG to the experiments.
- Trying the same experiments with different datasets to see if the results are dataset dependent.

- Extending the current end-to-end system by adding Siamese and Triplet networks and find a way to compare all 3 different models.
- Adding more modules to the current system and diversifying the search methods, and/or extending the system in a way that hashing/indexing is applied with or instead of dimensionality reduction.
- Adding different similarity matching libraries such as ANNOY, FAISS, and NMSLib and compare the results with them.
- Making experiments on different metrics such as cosine similarity, L1, L2 for pairwise distance measures.
- Extracting different types of features such as texture and patterns and combining with the previous features to see if the search relevance will be improved.
- Use the newly extracted features such as texture, patterns along with CNN features in Siamese and Triplet networks and perform multi-metric learning to optimized for all of the results. Try also by weighting the different feature types.

Appendix A

Appendix

A.1 Experiment 1 Additional Figures and Tables

Baseline Name	best_1	best_2	best_3	best_4	best_5
colorhistogram	lawn_mower : 2.14	orange : 1.9	sunflower : 1.64	shark : 1.45	cockroach : 1.42
colormoments	orange : 1.67	apple : 1.27	poppy : 1.01	cloud : 0.7	plain : 0.65
hog	plate : 5.36	lawn_mower : 3.75	can : 3.08	plain : 2.61	chair : 2.59
inceptionresnetv2	chair : 8.6	keyboard : 8.43	clock : 8.23	chimpanzee : 8.2	orange : 8.18
inceptionv3	chair : 8.51	keyboard : 8.12	orange : 7.93	chimpanzee : 7.9	bottle : 7.74
mobilenet	wardrobe : 8.21	apple : 7.45	chair : 7.06	television : 6.67	cockroach : 6.66
resnet50	cup : 2.96	sunflower : 2.72	can : 2.54	bottle : 2.42	dinosaur : 2.29
vgg16	wardrobe : 5.31	apple : 4.63	chair : 4.37	lawn_mower : 3.96	plain : 3.84
vgg19	caterpillar : 5.26	sunflower : 4.41	maple_tree : 4.38	cockroach : 4.31	elephant : 4.19
xception	chair : 8.31	keyboard : 8.28	wardrobe : 8.14	orange : 7.93	apple : 7.75

Table A.1: Experiment 1 Class@10 Best labels per Baseline

Baseline Name	worst_1	worst_2	worst_3	worst_4	worst_5
colorhistogram	lamp : 0.1	television : 0.12	man : 0.14	rabbit : 0.16	bed : 0.18
colormoments	television : 0.1	otter : 0.11	turtle : 0.11	flatfish : 0.12	mouse : 0.12
hog	man : 0.05	orchid : 0.07	turtle : 0.09	seal : 0.09	bee : 0.09
inceptionresnetv2	shrew : 2.21	otter : 2.26	girl : 2.27	crocodile : 2.37	boy : 2.63
inceptionv3	mouse : 1.66	otter : 1.68	shrew : 1.75	crocodile : 1.96	seal : 1.99
mobilenet	seal : 1.23	otter : 1.34	lizard : 1.35	caterpillar : 1.42	mouse : 1.5
resnet50	bus : 0.21	otter : 0.22	pickup_truck : 0.22	hamster : 0.25	road : 0.26
vgg16	woman : 0.38	seal : 0.39	otter : 0.51	snake : 0.51	bear : 0.53
vgg19	rabbit : 0.47	leopard : 0.49	hamster : 0.49	bus : 0.54	lobster : 0.55
xception	otter : 1.75	seal : 1.75	mouse : 1.8	crocodile : 1.93	beaver : 2.03

Table A.2: Experiment 1 Class@10 Worst labels per Baseline

Baseline Name	best_1	best_2	best_3	best_4	best_5
colorhistogram	flowers : 2.23	trees : 2.112	fruit_and_vegetables : 1.7	aquatic_mammals : 1.56	large_carnivores : 1.486
colormoments	flowers : 1.808	fruit_and_vegetables : 1.55	trees : 1.304	large_man-made_outdoor_things : 1.086	household_electrical_devices : 0.98
hog	food_containers : 3.848	large_natural_outdoor_scenes : 2.872	trees : 2.542	large_carnivores : 2.36	household_electrical_devices : 2.06
inceptionresnetv2	household_furniture : 8.364	food_containers : 8.178	fruit_and_vegetables : 7.988	household_electrical_devices : 7.96	trees : 7.85
inceptionv3	food_containers : 8.0	household_furniture : 7.926	trees : 7.888	fruit_and_vegetables : 7.61	household_electrical_devices : 7.52
mobilenet	trees : 7.52	large_natural_outdoor_scenes : 7.22	household_furniture : 6.548	fruit_and_vegetables : 6.312	flowers : 6.224
resnet50	food_containers : 3.74	people : 2.564	reptiles : 2.21	household_furniture : 2.174	trees : 2.064
vgg16	large_natural_outdoor_scenes : 4.604	trees : 4.5	household_furniture : 4.214	fruit_and_vegetables : 3.722	food_containers : 3.212
vgg19	food_containers : 4.71	people : 4.214	insects : 3.888	large_omnivores_and_herbivores : 3.566	trees : 3.536
xception	household_furniture : 7.872	food_containers : 7.816	household_electrical_devices : 7.768	fruit_and_vegetables : 7.722	trees : 7.7

Table A.3: Experiment 1 SuperClass@10 Best labels per Baseline

Baseline Name	worst_1	worst_2	worst_3	worst_4	worst_5
colorhistogram	people : 0.896	household_furniture : 0.904	large_omnivores_and_herbivores : 0.992	small_mammals : 0.996	household_electrical_devices : 1.032
colormoments	food_containers : 0.616	reptiles : 0.666	fish : 0.714	insects : 0.718	small_mammals : 0.728
hog	people : 0.262	large_omnivores_and_herbivores : 0.482	flowers : 0.652	large_man-made_outdoor_things : 0.74	insects : 0.89
inceptionresnetv2	reptiles : 4.908	non-insect_invertebrates : 5.312	small_mammals : 5.678	aquatic_mammals : 5.706	insects : 5.936
inceptionv3	reptiles : 4.19	non-insect_invertebrates : 4.544	small_mammals : 4.58	aquatic_mammals : 4.78	insects : 5.368
mobilenet	reptiles : 3.112	non-insect_invertebrates : 3.38	small_mammals : 3.538	medium_mammals : 3.85	insects : 4.05
resnet50	vehicles_1 : 0.998	large_man-made_outdoor_things : 1.074	non-insect_invertebrates : 1.114	fruit_and_vegetables : 1.186	small_mammals : 1.284
vgg16	non-insect_invertebrates : 1.46	reptiles : 1.636	small_mammals : 1.754	medium_mammals : 1.816	large_omnivores_and_herbivores : 1.892
vgg19	fruit_and_vegetables : 1.468	non-insect_invertebrates : 1.546	large_natural_outdoor_scenes : 1.63	large_man-made_outdoor_things : 1.734	large_carnivores : 1.75
xception	reptiles : 3.848	non-insect_invertebrates : 4.258	aquatic_mammals : 4.808	small_mammals : 4.87	fish : 4.962

Table A.4: Experiment 1 SuperClass@10 Worst labels per Baseline

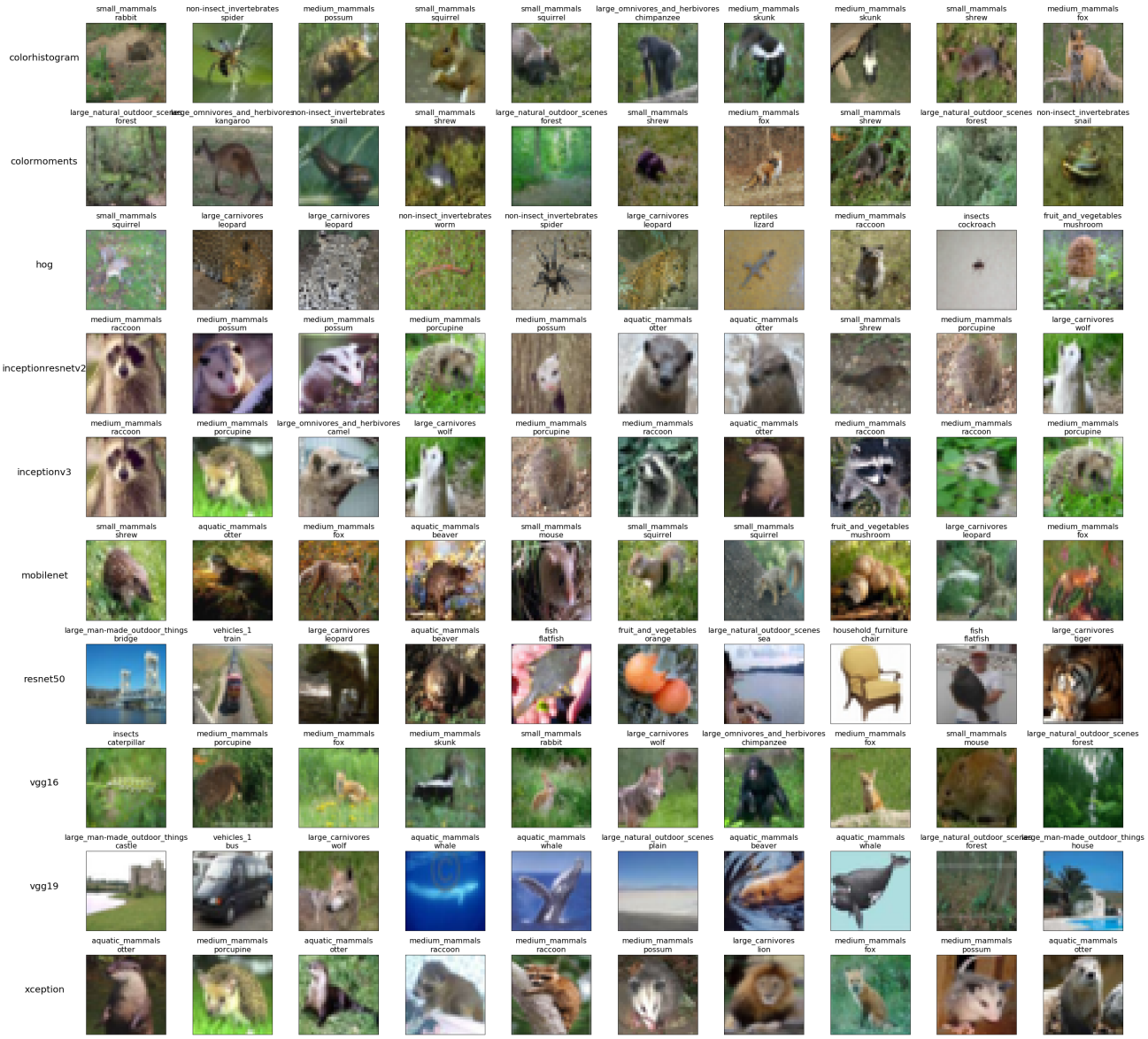

Figure A.1: Random Otter Image Query and Retrieved Similar Images per Baseline

A.2 Experiment 2 Additional Figures and Tables

Baseline Name	mean	std	min	25%	50%	75%	max
ch-hog	0.92	1.87	0.0	0.0	0.0	1.0	10.0
cm-ch	0.63	1.20	0.0	0.0	0.0	1.0	10.0
cm-ch-hog	0.93	1.88	0.0	0.0	0.0	1.0	10.0
cm-hog	0.91	1.87	0.0	0.0	0.0	1.0	10.0
inres2-cm-ch	5.32	3.68	0.0	2.0	5.0	9.0	10.0
inres2-cm-ch-hog	3.69	3.30	0.0	1.0	3.0	6.0	10.0
inres2-hog	3.67	3.30	0.0	1.0	3.0	6.0	10.0
inres2-inv3	5.45	3.68	0.0	2.0	6.0	9.0	10.0
inres2-mn	5.49	3.59	0.0	2.0	6.0	9.0	10.0
inres2-res50	5.28	3.66	0.0	2.0	5.0	9.0	10.0
inres2-vgg16	5.35	3.67	0.0	2.0	5.0	9.0	10.0
inres2-vgg19	5.35	3.67	0.0	2.0	5.0	9.0	10.0
inres2-xcp	5.43	3.65	0.0	2.0	5.0	9.0	10.0
mn-cm-ch	3.65	3.24	0.0	1.0	3.0	6.0	10.0
mn-cm-ch-hog	2.01	2.63	0.0	0.0	1.0	3.0	10.0
mn-hog	1.98	2.62	0.0	0.0	1.0	3.0	10.0

Table A.5: Class@10 Description Table for Experiment 2

Baseline Name	mean	std	min	25%	50%	75%	max
ch-hog	1.63	2.24	0.0	0.0	1.0	2.0	10.0
cm-ch	1.39	1.70	0.0	0.0	1.0	2.0	10.0
cm-ch-hog	1.64	2.25	0.0	0.0	1.0	2.0	10.0
cm-hog	1.62	2.24	0.0	0.0	1.0	2.0	10.0
inres2-cm-ch	6.92	3.46	0.0	4.0	9.0	10.0	10.0
inres2-cm-ch-hog	5.30	3.41	0.0	2.0	5.0	9.0	10.0
inres2-hog	5.27	3.40	0.0	2.0	5.0	9.0	10.0
inres2-inv3	7.08	3.41	0.0	4.0	9.0	10.0	10.0
inres2-mn	7.06	3.33	0.0	4.0	9.0	10.0	10.0
inres2-res50	6.91	3.45	0.0	4.0	8.5	10.0	10.0
inres2-vgg16	6.95	3.44	0.0	4.0	9.0	10.0	10.0
inres2-vgg19	6.95	3.44	0.0	4.0	9.0	10.0	10.0
inres2-xcp	7.02	3.41	0.0	4.0	9.0	10.0	10.0
mn-cm-ch	5.08	3.40	0.0	2.0	5.0	8.0	10.0
mn-cm-ch-hog	3.14	3.02	0.0	1.0	2.0	5.0	10.0
mn-hog	3.10	3.01	0.0	1.0	2.0	5.0	10.0

Table A.6: SuperClass@10 Description Table for Experiment 2

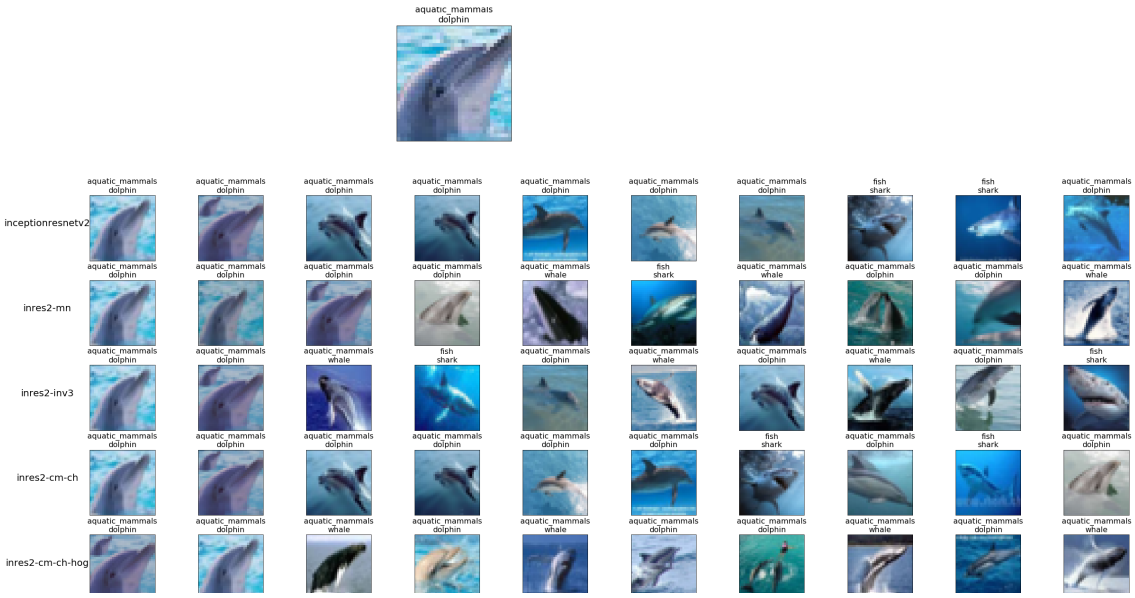

Figure A.2: Retrieved Results for Query Image

Figure A.3: Retrieved Results for Query Image

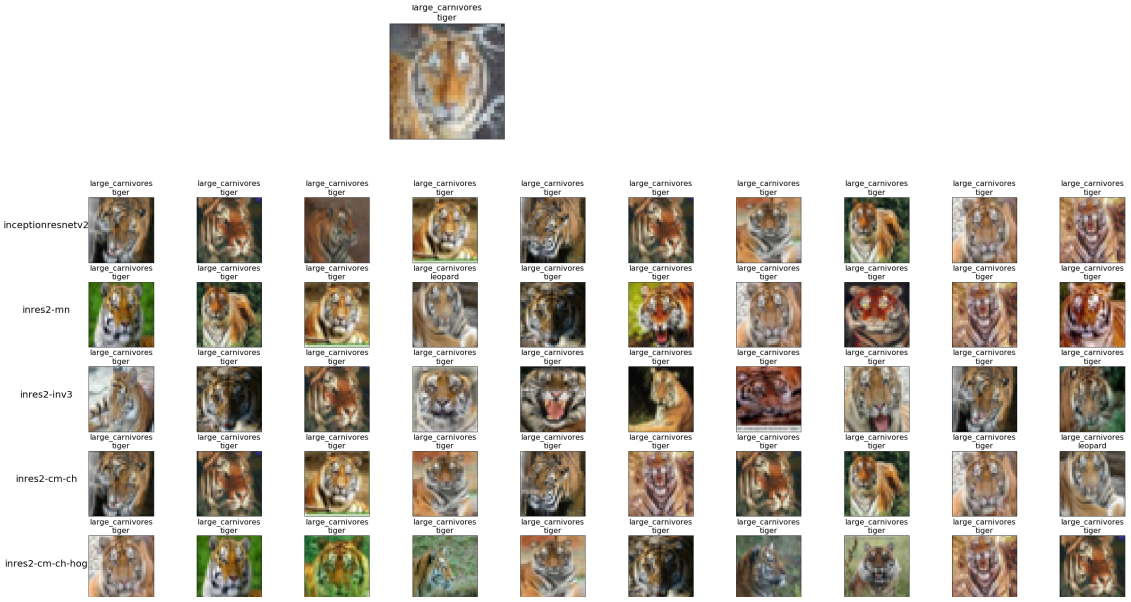

Figure A.4: Retrieved Results for Query Image

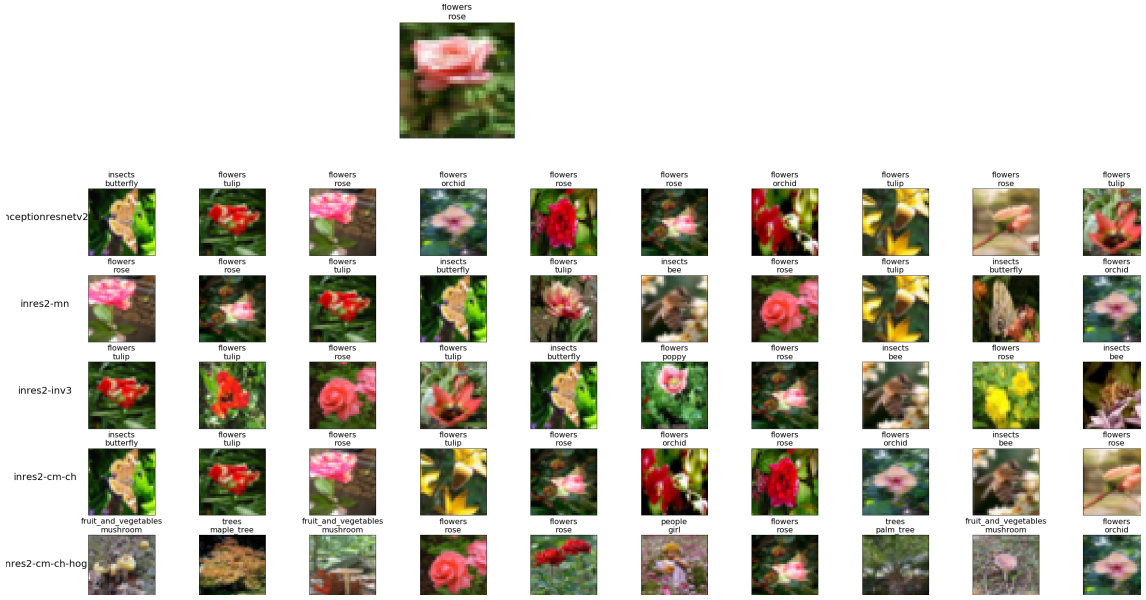

Figure A.5: Retrieved Results for Query Image

Figure A.6: Retrieved Results for Query Image

A.3 Experiment 3 Additional Figures and Tables

Baseline Name	mean	std	min	25%	50%	75%	max
inceptionresnetv2_lda_367	3.22	2.01	1.0	2.0	3.0	4.0	10.0
inceptionresnetv2_lda_528	3.22	2.01	1.0	2.0	3.0	4.0	10.0
inceptionresnetv2_lda_937	3.22	2.01	1.0	2.0	3.0	4.0	10.0
inceptionresnetv2_nca_367	6.51	1.87	1.0	5.0	7.0	8.0	10.0
inceptionresnetv2_nca_528	6.67	1.84	1.0	5.0	7.0	8.0	10.0
inceptionresnetv2_nca_937	6.84	1.82	1.0	6.0	7.0	8.0	10.0
inceptionresnetv2_pca_367	9.50	0.60	7.0	9.0	10.0	10.0	10.0
inceptionresnetv2_pca_528	9.68	0.49	7.0	9.0	10.0	10.0	10.0
inceptionresnetv2_pca_937	9.88	0.33	8.0	10.0	10.0	10.0	10.0
inceptionresnetv2_random_367	7.76	1.37	2.0	7.0	8.0	9.0	10.0
inceptionresnetv2_random_528	8.14	1.22	3.0	7.0	8.0	9.0	10.0
inceptionresnetv2_random_937	8.55	1.05	4.0	8.0	9.0	9.0	10.0
inceptionresnetv2_umap_367	4.01	1.76	1.0	3.0	4.0	5.0	10.0
inceptionresnetv2_umap_528	4.01	1.74	1.0	3.0	4.0	5.0	10.0
inceptionresnetv2_umap_937	3.99	1.74	1.0	3.0	4.0	5.0	10.0

Table A.7: Intersection@10 Description Table for Experiment 3

Baseline Name	mean	std	min	25%	50%	75%	max
inceptionresnetv2_lda_367	0.43	0.13	0.080556	0.327083	0.430556	0.538368	0.806250
inceptionresnetv2_lda_528	0.43	0.13	0.080556	0.327083	0.430556	0.538368	0.806250
inceptionresnetv2_lda_937	0.43	0.13	0.080556	0.327083	0.430556	0.538368	0.806250
inceptionresnetv2_nca_367	0.47	0.14	0.116667	0.360417	0.460417	0.575000	0.893750
inceptionresnetv2_nca_528	0.48	0.14	0.108333	0.366667	0.468750	0.579167	0.966667
inceptionresnetv2_nca_937	0.48	0.15	0.114583	0.366667	0.475000	0.588889	0.918750
inceptionresnetv2_pca_367	0.67	0.13	0.139583	0.577083	0.677431	0.764583	1.000000
inceptionresnetv2_pca_528	0.72	0.13	0.212500	0.637500	0.731250	0.812500	1.000000
inceptionresnetv2_pca_937	0.79	0.10	0.220833	0.725000	0.806250	0.862500	1.000000
inceptionresnetv2_random_367	0.50	0.15	0.100000	0.391667	0.495833	0.612500	0.968750
inceptionresnetv2_random_528	0.52	0.15	0.104167	0.406250	0.514583	0.627083	1.000000
inceptionresnetv2_random_937	0.55	0.15	0.105556	0.437500	0.541667	0.656250	1.000000
inceptionresnetv2_umap_367	0.42	0.12	0.066667	0.327083	0.416667	0.518750	0.818750
inceptionresnetv2_umap_528	0.42	0.12	0.075000	0.327083	0.416667	0.520833	0.868750
inceptionresnetv2_umap_937	0.42	0.12	0.091667	0.325000	0.416667	0.518750	0.787500

Table A.8: MAP@10 Description Table for Experiment 3

Figure A.7: Retrieved Results for Query Image

Figure A.8: Retrieved Results for Query Image

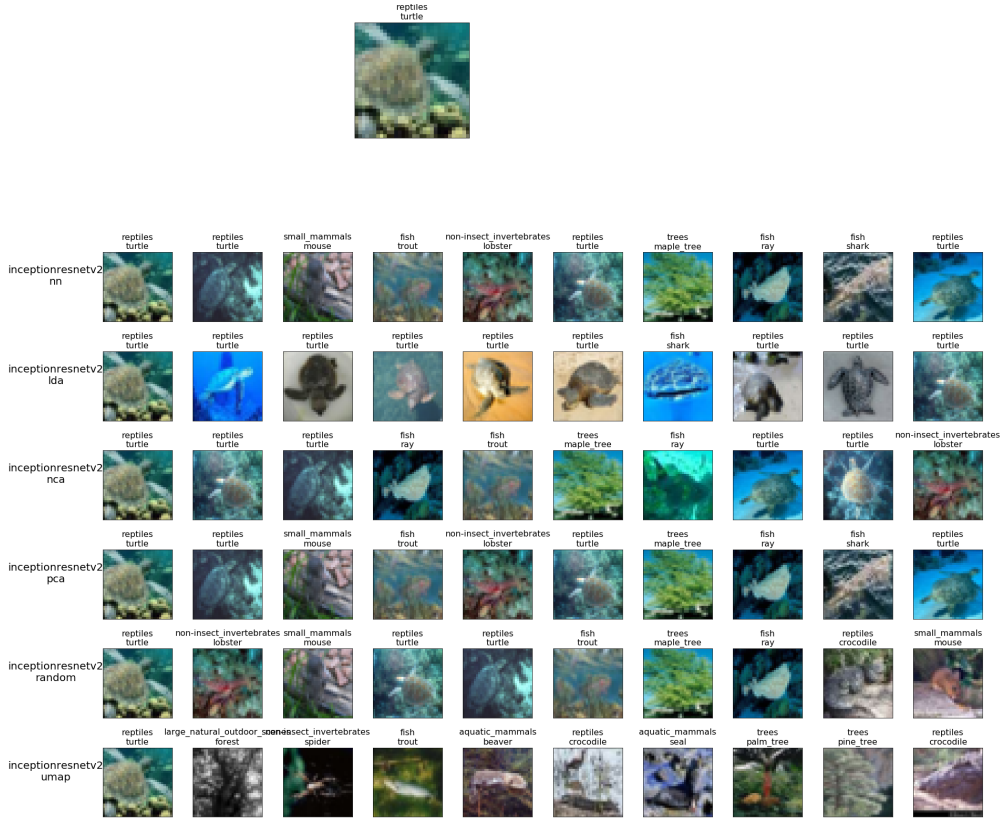

Figure A.9: Retrieved Results for Query Image

Baseline Name	Dimensions	Index Time	Search Time
inceptionresnetv2_lda_367	99	0.19	2.49
inceptionresnetv2_lda_528	99	0.18	2.26
inceptionresnetv2_lda_937	99	0.21	2.56
inceptionresnetv2_nca_367	367	0.43	2.97
inceptionresnetv2_nca_528	528	0.47	3.21
inceptionresnetv2_nca_937	937	0.94	4.77
inceptionresnetv2_pca_367	367	0.22	2.86
inceptionresnetv2_pca_528	528	0.24	2.98
inceptionresnetv2_pca_937	937	0.52	4.70
inceptionresnetv2_random_367	367	0.20	2.85
inceptionresnetv2_random_528	528	0.25	2.97
inceptionresnetv2_random_937	937	0.49	4.93
inceptionresnetv2_umap_367	367	0.13	3.23
inceptionresnetv2_umap_528	528	0.11	3.49
inceptionresnetv2_umap_937	937	0.12	3.93

Table A.9: Index and Search Time Results for all models in Experiment 3

Bibliography

[1] Srikar Appalaraju and Vineet Chaoji. Image similarity using deep cnn and curriculum learning. *arXiv preprint arXiv:1709.08761*, 2017.

[2] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18:1–8, 1998.

[3] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

[4] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[5] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *CoRR*, abs/1605.07678, 2016.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. 2005.

[7] Sanjoy Dasgupta. Experiments with random projection. *arXiv preprint arXiv:1301.3849*, 2013.

[8] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[9] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008.

[10] Robert Geirhos, David HJ Janssen, Heiko H Schütt, Jonas Rauber, Matthias Bethge, and Felix A Wichmann. Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*, 2017.

[11] Jacob Goldberger, Geoffrey E Hinton, Sam T Roweis, and Ruslan R Salakhutdinov. Neighbourhood components analysis. In *Advances in neural information processing systems*, pages 513–520, 2005.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[13] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[14] Heike Hofmann, Karen Kafadar, and Hadley Wickham. Letter-value plots: Boxplots for large data. Technical report, had.co.nz, 2011.

[15] Houdong Hu, Yan Wang, Linjun Yang, Pavel Komlev, Li Huang, Xi Stephen Chen, Jiapei Huang, Ye Wu, Meenaz Merchant, and Arun Sacheti. Web-scale responsive visual search at bing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 359–367. ACM, 2018.

[16] Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, Jiajing Xu, Jeff Donahue, and Sarah Tavel. Visual search at pinterest. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1889–1898. ACM, 2015.

[17] Noah Keen. Color moments. *School Of Informatics, University Of Edinburgh*, pages 3–6, 2005.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] Vijay Kumar and Priyanka Gupta. Importance of statistical measures in digital image processing. *International Journal of Emerging Technology and Advanced Engineering*, 2(8):56–62, 2012.

[21] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

[23] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. Siamese network features for image matching. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 378–383. IEEE, 2016.

[24] Cun Mu, Jun Zhao, Guang Yang, Jing Zhang, and Zheng Yan. Towards practical visual search engine within elasticsearch. *arXiv preprint arXiv:1806.08896*, 2018.

[25] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[26] Sebastian Raschka. Stat 479: Machine learning lecture notes, 2018.

[27] Josef Sivic and Andrew Zisserman. Video google: Efficient visual search of videos. In *Toward category-level object recognition*, pages 127–144. Springer, 2006.

[28] SR Surya and G Sasikala. Survey on content based image retrieval. *Indian J. Comput. Sci. Eng*, 2:691–696, 2011.

[29] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[30] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[31] Fan Yang, Ajinkya Kale, Yury Bubnov, Leon Stein, Qiaosong Wang, Hadi Kiapour, and Robinson Piramuthu. Visual search at ebay. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2101–2110. ACM, 2017.

[32] Xinpan Yuan, Qunfeng Liu, Jun Long, Lei Hu, and Yulou Wang. Deep image similarity measurement based on the improved triplet network with spatial pyramid pooling. *Information*, 10(4):129, 2019.

[33] Andrew Zhai, Dmitry Kislyuk, Yushi Jing, Michael Feng, Eric Tzeng, Jeff Donahue, Yue Li Du, and Trevor Darrell. Visual discovery at pinterest. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 515–524. International World Wide Web Conferences Steering Committee, 2017.

[34] Andrew Zhai, Hao-Yu Wu, Eric Tzeng, Dong Huk Park, and Charles Rosenberg. Learning a unified embedding for visual search at pinterest. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2412–2420, New York, NY, USA, 2019. ACM.

[35] Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, and Rong Jin. Visual search at alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 993–1001. ACM, 2018.

[36] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR 2011*, pages 809–816. IEEE, 2011.