# Methods on Cross-modal Retrieval with Metric Learning

Igor Ugarte Molinet
igorugarte.cvm@gmail.com

Juan Antonio Rodríguez García
juanantonio.rodriguez@upf.edu

Francesc Net Barnès
francescnet@gmail.com

David Serrano Lozano
99d.serrano@gmail.com

Master in Computer Vision
Universitat Autonima de Barcelona
Barcelona, Spain

## Abstract

*Image retrieval task is based on finding a set of k most similar images in a database given some query image. This task is mostly done in a brute-force scheme, by using k-Nearest Neighbors algorithm. An efficient and compact strategy to encode the images in a feature embedding is needed for both storing the data and performing computations. The performance of a Image Retrieval system is inherently constrained by the features adopted to represent the images, both in the query set and in the database. This work presents some methods for metric learning of feature embeddings that are good for image retrieval in the MIT-Scene dataset. Furthermore, we explore the problem of cross-modal retrieval, where image and text pairs are projected to a common space where distances can be computed and a retrieval algorithm can be applied. We present results for different types of embeddings, specifically VGG and Faster RCNN for images, and FastText and BERT for texts. https://github.com/cesc47/cross-modal-retrieval-with-triplet-network*

## 1. INTRODUCTION

The objective of Image retrieval [2] is to return similar images from a database with respect to a given query image. A similarity measure must be used to quantify how similar are two images, and a feature embedding strategy must be designed to effectively discern between images. The most used similarity measures through the literature have been the Euclidean L2 distance or the Mahalanobis distance, but they introduce a hard bias to a certain coordinate space. A more robust approach, now in that deep learning is able to model non-linear patterns, would be to also learn a similarity measure through Metric learning. Furthermore, Metric learning through Siamese or Triplet networks, based on CNN, will allow models to learn well suited feature embeddings for the task.

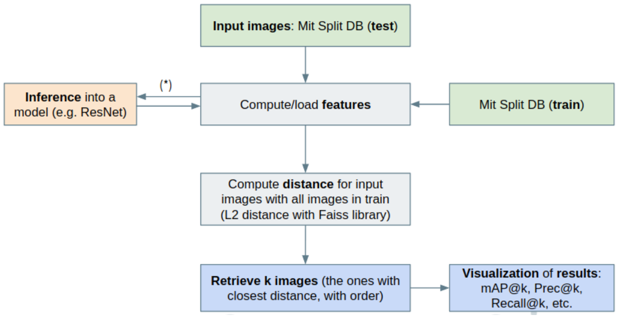

Input images: Mit Split DB (test)
Inference into a model (e.g. ResNet)
(*)
Compute/load features
Mit Split DB (train)
Compute distance for input images with all images in train (L2 distance with Faiss library)
Retrieve k images (the ones with closest distance, with order)
Visualization of results: mAP@k, Prec@k, Recall@k, etc.

Figure 1. Pipeline of an Image retrieval system.

In the second part of this work we explore the cross-modal retrieval scenario. For this task, a pair of images and captions are presented to the architecture, and the training procedure manages to project features to the same space where similar instances lay are close and dissimilar pairs are far. That is, extract meaningful features from both images and texts using clever approaches such as VGG or Faster RCNN for images and FastText and BERT embeddings for texts. Then, all features are projected to the same cross-modal space, so that image and text embeddings are vectors in the same space, and similarity distances are imposed thanks to a Triplet metric scheme.

This paper is structured as follows: Section 2 presents the methods used through the literature for image retrieval and metric learning, Section 3 introduces our methodologies, Section 4 presents datasets and experimental results, and finally Section 5 we draw conclusions.

## 2. RELATED WORK

In this section we explain the different methods that have been proposed the last decade for image retrieval.

### 2.1. Handcrafted methods

Historically, there has been a lot of methods in order to extract the features of the images in order to compare them and perform the Content Based Image Retrieval (CBIR). Some of the examples could be color, texture, shape, gradient, etc. This characteristics are represented in a form of a feature vector, and it is compared taking account a specific distance metric to get the similarity results.

### 2.2. ML and DL based

Nowadays due to the technological advances in computing, is typical to use deep learning in order to obtain the feature vectors or to generate the retrievals. A lot of techniques can be performed: the type of supervision can be changed, the network, the type of retrieval, etc.

## 3. Methods

Before explaining the methods is important to define the Contrastive Loss, and the Triplet Loss that will be used in the Siamese and Triplet networks, respectively:

The **Contrastive loss** gets the output of the network for a positive example and calculates its distance to an example of the same class and contrasts that with the distance to negative examples. This means that ideally the positive samples should be encoded in closer representations and the negative samples (different classes) should be encoded in further representations in order to have a low loss.

The **Triplet loss** is a function that tries to minimise the distance between a baseline (also called anchor image) and a positive sample (from the same class as the anchor) and to maximise the distance between the baseline and a negative sample.

### 3.1. Retrieval with a pre-trained NN architecture

A pre-trained architecture is used to generate the feature vectors. In this paper it will be used the family of ResNet [3], specifically ResNet18, ResNet50 and ResNet101.

### 3.2. Siamese Networks

It is a class of Neural Network architecture that contains two identical networks, i.e. same model and weights. The key on a Siamese Network is to find the similarity of the inputs by comparing its feature vectors. In this paper, the Resnet family of networks is used.

The workflow of this networks is the following: Two images are taken randomly, which could be of the same class (annotated as similar) or from a different class (annotated as dissimilar). Then and a feature embedding is generated, and the contrastive loss is used to learn how to differentiate both scenarios. When the training converges, a well-suited set of weights are obtained for retrieval and classification.

### 3.3. Triplet Networks

The Triplet network [4] was inspired by the Siamese network, and tries to tackle the problem of using easy instances while training, that is, showing pairs of images that are good to classify. This network uses 3 instances of the same feed-forward network, with shared parameters. Three images will be provided to the model, where two of them will be similar (anchor and positive samples), and the third will be dissimilar (a negative example). As it has more rich information in each training instance, this strategy promises faster convergence and better class separability.

### 3.4. Cross-modal retrieval using Triplet metric learning

For the second part of this work we employ the Triplet metric learning scheme to learn a joint distribution for texts and images, that will allow our system to retrieve text captions from images and vice-versa. This method projects image and text embeddings to a cross-modal space where distances between vectors can be computed, and KNN retrieval is performed. To this end, we experiment with VGG and Faster RCNN image features and FastText and BERT [1] embeddings for the texts.

## 4. Experimental results

This section is devoted to draw some experiments with the described methods, and present both quantitative and qualitative results.

### 4.1. Mit split dataset

The MIT-Scene dataset consists of 2688 squared images of 256 pixels per side. The images consist in natural scenes divided in 8 types: coast, forest, highway, inside city, mountain, opencountry, street and tallbuilding. One thing to take into account is that the number of instances per class is quite similar thus we can consider that we do not have a class imbalance problem. This dataset has shown to present much confusion over classes even for humans, as there are examples that could be associated with many classes.

### 4.2. Flickr30 dataset

The Flickr30 dataset [5] consists of 31.783 images and 158.915 text captions (5 captions for each image, describing it with different words but simmilar meaning). We apply the original splits to the dataset for train, test and validation. Specifically, we use 29000 for train, 1000 for validation and 1014 for test. The images in the dataset consist on in-the-wild samples, showing events and scenes with a variety of conditions. The captions are annotated via crowd-sourcing.

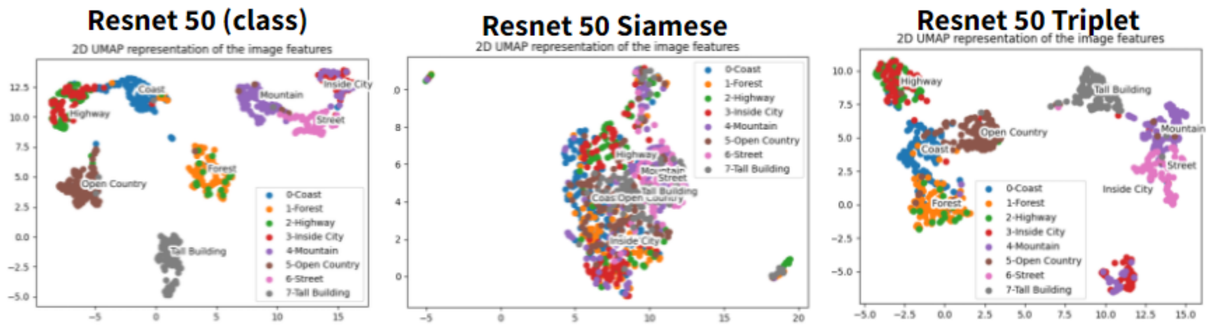


Figure 2. U-MAP comparison between Resnet 50 trained for classification on MIT (left), Siamese Resnet 50 (middle) and Triplet Resnet 50 (right)

| Model | prec@1 | prec@5 | map@1 | map@5 |
|---|---|---|---|---|
| Resnet18 | 0.3500 | 0.8100 | 0.3200 | 0.3700 |
| Resnet50 | 0.6700 | 0.9000 | 0.6200 | 0.7200 |
| Resnet101 | 0.6800 | 0.9000 | 0.6400 | 0.7300 |

Table 1. Results for image retrieval on Resnet architectures, using pretrained weigths on Imagenet

### 4.3. Evaluation Metrics

The performance of the different learning models created has been analyzed with the following metrics:

#### 4.3.1 MAP@k

This metric represents the mean average precision at K i.e. look for a correct retrieval in the first k candidates. To evaluate the retrieval system, we average the precision computed for each query and the mean over all queries.

$$mAP@k = \frac{1}{N}\sum_{i=1}^{N} AP@k_i \tag{1}$$

where

$$AP@k = \frac{1}{k}\sum_{i=1}^{k} P@i, \tag{2}$$

### 4.4. Pretrained Resnet of Imagenet

In this experiment we aim to asses how good are Imagenet weights of Resnet architectures at generating embeddings for the task of image retrieval. We test Resnet 18, Resnet 50 and Resnet 101. Results are depicted in table 1, showing similar performance of Resnet 50 and 101, clearly outperforming Resnet 18.

We also show more qualitative results in figure 3, where the clustering is plotted using PCA and U-MAP for Resnet 18 and 101. This plots show how the different classes are mixed in the 2D projection, while we would need better separability. Resnet 101 and U-MAP do a better job at distinguishing the different classes, compared to PCA and Resnet 18.

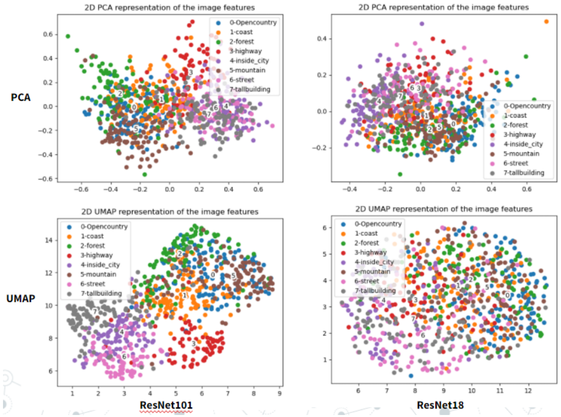


Figure 3. PCA and U-MAP project the feature embeddings into the 2D space to assess the clustering of Resnet 18 and 101 pretrained with Imagenet.

### 4.5. Fine-tuning models with MIT-Scene

In this experiment we test Resnet 50 and a custom shallow CNN architecture for generating separable feature embeddings and testing the image retrieval. We stick to Resnet 50 as it gave the best trade-off between accuracy and speed. Our custom CNN (P1 CNN) is based on two blocks of convolutions, batch norm. and dropout, which aims to be very compact. We train both models with categorical cross-entropy loss for classification. We use Adam optimizer, LR of 0.001 (scheduled) batches of 32, 50 epochs, and normalize using Imagenet mean and std. Quantitative results are presented in table 2, and show how Resnet 50 outperforms our shallow CNN both in mAP@k and prec@k. We also show qualitative results in figure 4, where we can observe that Resnet 50 generates separable classes, while P1 CNN

| Model | prec@1 | prec@5 | map@1 | map@5 |
|---|---|---|---|---|
| Resnet50 | 0.7707 | 0.9504 | 0.7640 | 0.8480 |
| P1 CNN | 0.6023 | 0.8958 | 0.6045 | 0.7205 |

Table 2. Results for image retrieval on two CNN based methods, trained with MIT-Scene

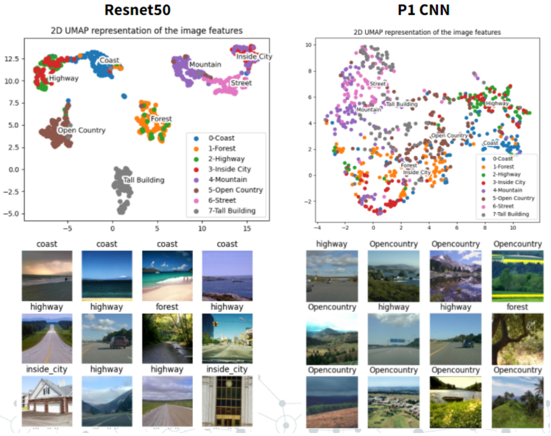


Figure 4. Cluster 2D projection using U-MAP, for Resnet 50 and P1 CNN trained in MIT-Scene. In the images section, the first column corresponds to queries, and the other are ordered retrievals, and each row is a query

| Model | prec@1 | prec@5 | map@1 | map@5 |
|---|---|---|---|---|
| Resnet50 | 0.5718 | 0.8669 | 0.5724 | 0.6891 |
| Resnet 50+FC 2 | 0.4466 | 0.8671 | 0.4448 | 0.6157 |

Table 3. Results for image retrieval in a Siamese-based strategy, using Resnet 50 and two types of heads, on MIT-Scene

has more problems and mixes the classes. Furthermore, we can observe that the retrievals are more robust in Resnet 50, by comparing the query with the retrievals.

### 4.6. Siamese metric learning

We perform Siamese learning in a Resnet 50 backbone, using the same training setup described before. We test two different classification heads to test the method: The first one (Resnet 50) is just removing the linear layer of the Resnet 50, which turns out to be the Average pooling layer of size 2048. The second one (Resnet 50+FC 2) is to add a layer of MLP and PRelu, to encode the 2048 feature vector into a size 2 vector. This second strategy will allow us to learn a good 2D projection for both visualization and retrieval. Results of image retrieval are shown in table 3, where the Resnet 50 model outperforms the FC2 version. This is because, the FC2 encoding has oversimplified the embedding for retrieval and classification.

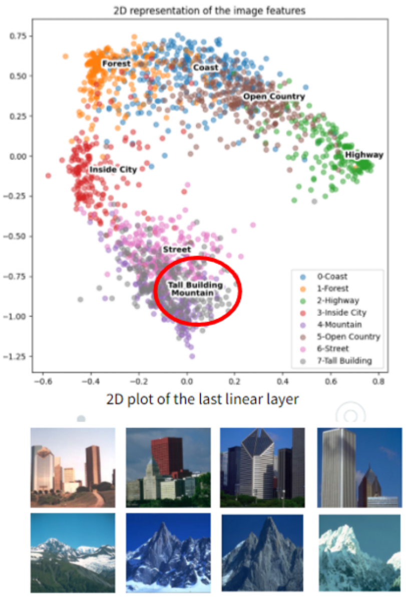


Figure 5. Qualitative results on the FC2 projection for the Siamese learning strategy. Tall building and mountain are clustered in the same region because they share the same spatial structure, large and tall structures with a clear sky in the background.

| Model | prec@1 | prec@5 | map@1 | map@5 |
|---|---|---|---|---|
| Resnet50 | 0.7353 | 0.9529 | 0.7447 | 0.8286 |
| Resnet 50+FC 2 | 0.5576 | 0.8899 | 0.5576 | 0.6895 |

Table 4. Results for image retrieval in a Triplet-based strategy, using Resnet 50 and two types of heads, on MIT-Scene

In figure 5 we observe the visualization of the FC2 projection resulting from the Siamese learning in MIT-Scene. It is clear that the classes are now separated according to the contrastive loss margin of 1, generating a setting similar to an ellipse. We can observe that some classes are missplaced, like Mountain and Tall Building. We attribute this phenomena to the fact that both examples show similar spatial structure, as they are large structures with a clear sky in the background.

### 4.7. Triplet metric learning

We perform Triplet learning in the Resnet 50 backbone, using the same training setup described before. Our methodology is the same as in the Siamese experiment, using the two types of heads defined. Table 4 depicts results for the Triplet scheme in the retrieval process, where also Resnet 50 is outperforming the FC2 version/

In figure 6 we present the clustering of the Resnet 50 with FC2 head, trained with the Triplet scheme. In comparison with the visualization of the Siamese, in figure 5,

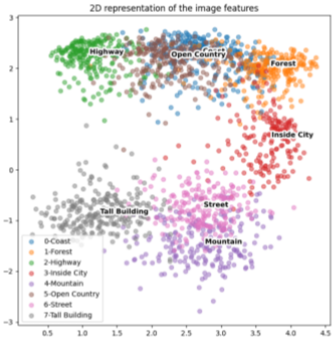


Figure 6. Cluster visualization of the Triplet learning with FC2 head. We can see how the classes that where mixed in the Siamese case now are separated, e.g. Mountain and Tall Building

now the classes of Mountain and Tall Building are well separated, because the training process was showing positive and negative examples, hence harder.

We can see in figure 2 a comparison of the U-MAP clustering visualization for the same Resnet 50 backbone, all of them trained in MIT-Scene. We can observe that the Triplet strategy is able to yield well separated classes, similar to the classification training. In contrast, Siamese learning mixes all classes, due to the fact that it did not find enough hard examples.

## 4.8. Cross Modal Retrieval

The objective of the Cross Modal Retrieval is to capture the semantic relationship between images and texts, and thanks to that, to create a system that is capable to extract an image (or texts) from a dataset given a caption (or image). The problem is posed as a Triplet metric learning, where the dataset instances are created with an anchor text (or image) and positive and negative text (or image), depending if the task is text-to-image or image-to-text retrieval.

All the experiments follow the same methodology, where image features and text features are extracted and projected to a common dense space. The triplet loss with margin will make dissimilar samples be at a distance higher or equal than the defined margin, 1 in our experiments. When the model is trained, the different data modalities will be projected to the same space, hence being able to to compute distance metrics between vectors.

The experimental setup is the following: Adam optimizer with initial learning rate or 0.0001 and exponential decay. We use a batch size of 1024, thanks to the use of precomputed features, during 50 epochs.

### 4.8.1 VGG and FastText and BERT features

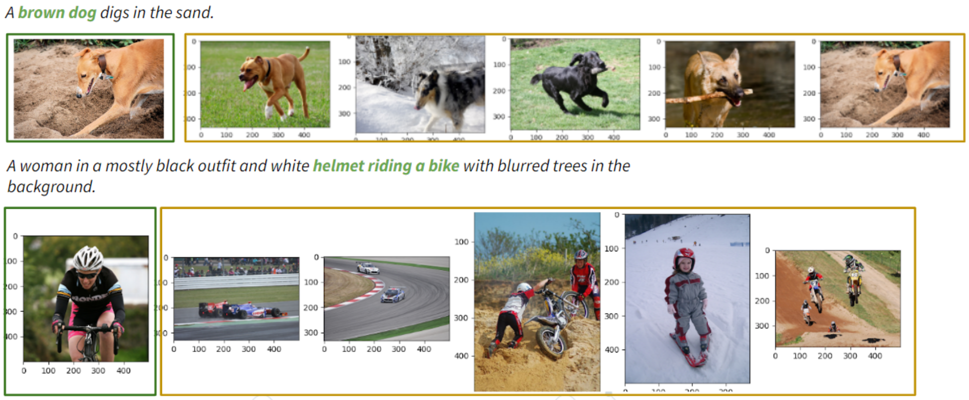
| Task | Model | map@1 | map@5 |
|---|---|---|---|
| Img-to-text | VGG+FastText(mean) | 0.052 | 0.013 |
| Img-to-text | VGG+FastText(sum) | 0.029 | 0.059 |
| Img-to-text | VGG+BERT | 0.026 | 0.087 |
| Text-to-Img | VGG+FastText(mean) | 0.010 | 0.018 |
| Text-to-Img | VGG+FastText(sum) | 0.018 | 0.025 |
| Text-to-Img | VGG+BERT | 0.023 | 0.036 |

Table 5. Results for the cross-modal retrieval using VGG + Fast-Text or Bert configuration.

In this case the feature extraction of images is done with VGG, a very dense CNN architecture capable of capturing rich spacial features. VGG features are projected to a dimension of 4096 using a linear layer to the cross-modal space. For the text embeddings we experiment with FastText and BERT embeddings. For the case of FastText, we employ mean and sum aggregation strategies, to flatten the embedding to a 1D feature array of size 300. In the case of BERT, we keep the feature embedding corresponding to the CLS toke, that is designed to capture information from the whole sentence. Finally, this features are projected to the 4096 space with a linear layer. Triplet loss is computed with the features obtained at the projected cross-modal space. As it can be seen in figure 8, results make semantic sense, where the high level context is captured. However, the fine-grained details in the sentences are not well obtained due to the coarse textual aggregation and the limitation of the VGG features. In figure 7 we can observer that the text-to-image scenario also behaves coherently, but it is not capable to retrieve the correct image. In table 5 we present a comparison between the different cross-modal methods using VGG. It is shown that the mAP@k metric is very poor due to the fact that there are a lot of plausible candidate retrievals and the model is not able to properly separate the samples. In the case of Text-to-Image, the problem is harder because there is only one correct instance, whereas in the Image-to-Text there are 5 possible correct instances.

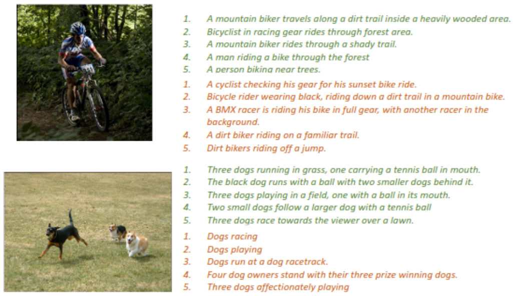
### 4.8.2 FasterRCNN with FastText and BERT features

In this experiment we aim to increase the spatial attention by using a object detection model. We make use of Faster R-CNN model pretrained in MS-COCO dataset, to extract features of Flickr30k dataset. This model returns a number of features that are aggregated using a mean over the final channels, hence projecting the features to a 1D space of 1024. Furthermore, we employ the same linear projection to the cross-modal space of 4096 dimensions. We use Detectron2 implementation. FastText with mean aggregation and BERT embeddings are used for encoding the text. For this experiments we stick to the Text-to-Image task. In tale 6 we present the results of this experiment. It is clear that results are unsatisfactory. We attribute this to the fact that we


A **brown dog** digs in the sand.

A woman in a mostly black outfit and white **helmet riding a bike** with blurred trees in the background.

Figure 7. Qualitative results of the Text-to-Image retrieval using VGG and FastText (mean) embeddings. The first image is the ground truth, and the 5 subsequent are the first 5 retrievals.


1. A mountain biker travels along a dirt trail inside a heavily wooded area.
2. Bicyclist in racing gear rides through forest area.
3. A mountain biker rides through a shady trail.
4. A man riding a bike through the forest
5. A person biking near trees.

1. A cyclist checking his gear for his sunset bike ride.
2. Bicycle rider wearing black, riding down a dirt trail in a mountain bike.
3. A BMX racer is riding his bike in full gear, with another racer in the background.
4. A dirt biker riding on a familiar trail.
5. Dirt bikers riding off a jump.

1. Three dogs running in grass, one carrying a tennis ball in mouth.
2. The black dog runs with a ball with two smaller dogs behind it.
3. Three dogs playing in a field, one with a ball in its mouth.
4. Two small dogs follow a larger dog with a tennis ball
5. Three dogs race towards the viewer over a lawn.

1. Dogs racing
2. Dogs playing
3. Dogs run at a dog racetrack.
4. Four dog owners stand with their three prize winning dogs.
5. Three dogs affectionately playing

Figure 8. Example of qualitative examples for the VGG + FastText configuration in the Cross Modal Retrieval for image-to-text. In green, the ground truth captions, and in orange the predictions of the network.

| Model | map@1 | map@5 |
|---|---|---|
| Faster RCNN+FastText(mean) | 0.001 | 0.002 |
| Faster RCNN+BERT | 0.001 | 0.002 |

Table 6. Results for the cross-modal Text-to-Image retrieval using Faster RCNN + FastText or Bert configuration.

are aggregating Faster RCNN features with a very coarse mean strategy, losing much information. We observed during training that the triplet loss rapidly converges to the margin value, which means that all instances are set to the same point and the model acts randomly.

## 5. Conclusions

We tested Resnet backbones and P1 CNN for Image retrieval and Metric learning. We evaluated pretrains with Imagenet and MIT, computed mAP@k and Prec@k, and visualized results with PCA, t-SNE, U-MAP and a learned FC 2 projection. We concluded that Resnet 50 is the best backbone, and we stick to U-MAP and FC 2 for visualization. FC 2 is preferable for visualization, as uses past layers to learn a 2D projection and shows both separation and transitions.

It is difficult to evaluate the model for classification when it was trained to differentiate images (Siamese or Triplet). We have shown that MIT-Scene have much confusing examples, resulting in mixed clusters. The triplet scheme is able to enforce hard examples, whereas Siamese may use a lot of easy examples.

For the cross-modal retrieval task using Flickr30k dataset, quantitative results are rather poor because the method is not able to catch the details in the image and text semantics. Also there are a lot of candidate retrievals that are plausible. Qualitative results with VGG and using BERT or FastText are more satisfactory, where high level concepts are captured, but details for a good retrieval are lost.

## References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[2] Shiv Ram Dubey. A decade survey of content based image retrieval using deep learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[4] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015. 2

[5] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 2