

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

Theses

---

10-2019

### Vector Spaces for Multiple Modal Embeddings

Sabarish Gopalakrishnan  
sxo8458@rit.edu

Follow this and additional works at: <https://repository.rit.edu/theses>

---

#### Recommended Citation

Gopalakrishnan, Sabarish, "Vector Spaces for Multiple Modal Embeddings" (2019). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

# Vector Spaces for Multiple Modal Embeddings

By

Sabarish Gopalakrishnan

October 2019

A Thesis Submitted  
in Partial Fulfillment  
of the Requirements for the Degree of  
Master of Science  
in  
Computer Engineering

Committee Approval:

---

Dr. Raymond Ptucha, *Advisor*  
Associate Professor

Date

---

Dr. Alexander Loui, *Committee Member*  
Professor

Date

---

Dr. Qi Yu, *Committee Member*  
Associate Professor

Date



Department of Computer Engineering

---

## Acknowledgments

I would like to take this opportunity to thank my advisor Dr. Raymond Ptucha for his continual support and guidance during my master's degree. I am thankful for Dr. Alexander Loui and Dr. Qi Yu for being in my thesis committee. I would like to give a personal thank you to Dr. Shagan Sah for being a mentor and guiding force throughout my master's. I would also like to thank my family and my close friends, without their support this journey would not have been as joyful as it was.

# Abstract

Deep learning has enabled great advances in the field of natural language processing, computer vision and pattern recognition in general. Deep learning frameworks have been very successful in performing classification, object detection, segmentation and translation. Before objects can be processed, a vector representation of that object needs to be created. For example, sentences and images can be encoded with a `sent2vec` and `image2vec` function respectively in preparation for input to a machine learning framework. Neural networks are able to learn efficient vector representation of images, text, audio, videos and 3D point clouds. However, the transfer of knowledge from one modality to the other is a challenging task. In this work, we develop vector spaces that can handle data that belongs to multiple modalities at the same time. In these spaces, similar objects are tightly clustered and dissimilar objects are far away irrespective of their modality. Such a vector space can be used in retrieval of objects, searching and generation tasks. For example, given a picture of a person surfing, one can retrieve sentences or audio bites of a person surfing. We build a Multi-stage Common Vector Space (M-CVS) and Reference Vector Space (RVS) that can handle images, text, audios, videos and 3D point cloud data. Both, the M-CVS and RVS can handle the addition of a new modality without having to change the existing transforms or architecture. Our model is evaluated by performing cross modal retrieval on multiple benchmark datasets.

# Contents

---

Vector Spaces for Multiple Modal Embeddings .....	1
List of Figures .....	6
List of Tables .....	8
Chapter 1 .....	9
1.1 Introduction .....	9
1.2 Motivation .....	10
1.3 Contributions .....	11
Chapter 2 .....	12
2.1 Deep Learning .....	12
2.2 Convolutional Neural Networks .....	12
2.3 Recurrent Neural Networks .....	14
A. Long Short Term Memory (LSTM) .....	16
B. Gated Recurrent United (GRU) .....	17
2.4 Multi-modal and Cross-modal Retrieval .....	18
2.5 Metric Learning Loss Functions .....	21
A. Contrastive Loss Function [38] .....	21
B. Triplet Loss Function [12] .....	21
C. Adversarial Loss Function [57] .....	22
2.6 Semi-Hard Negative Mining .....	23
Chapter 3 .....	24
3.1 Common Vector Space (CVS) .....	25
A. Loss Functions .....	27
B. Training Strategy .....	29
C. Aligned Attention .....	30
D. Limitations of the CVS Model .....	32
3.2 Multi-Staged Common Vector Space (M-CVS) .....	32
A. Training Strategy .....	33
3.3 Reference Vector Space (RVS) .....	34
A. Aligned Attention for RVS .....	35
3.4 Stage Wise Learning for Multi Modal Embeddings .....	37
A. Training Strategy .....	38
Chapter 4 .....	39

4.1	Datasets .....	39
A.	XMedia [11, 47] and XMediaNet [51,59].....	39
B.	Nuswide [49] and Pascal [50] .....	42
C.	Birds [58] .....	42
4.2	Implementation .....	43
A.	Evaluation Metric.....	43
B.	Intuitive Understanding of mAP .....	44
Chapter 5	.....	46
5.1	Results.....	46
A.	CVS Model: Pascal and Nuswide .....	46
B.	CVS Model: XMedia and XMediaNet (2 modalities) .....	47
C.	CVS Model: Xmedia and XMediaNet (5 modalities).....	47
D.	M-CVS Model: Xmedia and XMediaNet (5 modalities).....	49
E.	RVS Model: Xmedia and XMediaNet (5 modalities).....	50
F.	Stagewise Learning .....	52
5.2	Ablation Analysis.....	53
A.	Attention Ablation Analysis.....	53
B.	Timing Analysis.....	55
C.	Adversarial Loss Functions.....	55
Chapter 6	.....	57
6.1	Conclusions.....	57
6.2	Discussions .....	57
6.3	Future Work.....	58
Bibliography	.....	59

# List of Figures

---

Figure 1: Typical CNN architecture. An image passes through multiple convolutions and filtering operations before being passed through fully connected layers and classified into a one out of several classes. ....	13
Figure 2: Convolution Operation. Left – Input array; right – convolution kernel. ....	13
Figure 3: Max pooling operation in a convolutional neural network. ....	14
Figure 4: A recurrent neural network consisting of many sequential cells that take input from the previous state and also the current input to predict the output at each time step. ....	15
Figure 5: LSTM cell with input, output and forget cells that allows it to retain important information. ...	16
Figure 6: A GRU unit which has comparatively fewer parameters and is easier to train and tune. ....	17
Figure 7: The DSCMR [43] architecture consisting of three separate loss functions to perform cross modal retrieval. ....	20
Figure 8 Cases for the triplet loss. Loss is only updated in the bottom two cases when the negative is closer to the positive and the anchor. ....	22
Figure 9: The vector space before (left) and after (right) training. The colors represent the different data modalities and the shapes represent the categories. ....	24
Figure 10 Architecture for the CVS model. Data belonging to different modalities will get encoded into vectors after which they are projected into the common embedding space using fully connected layers. .	26
Figure 11 Cosine distance between two vectors in a 2D representation. ....	27
Figure 12: The adversarial loss function is calculated by comparing the generated and input images and sentences. ....	28
Figure 13 Aligned attention block which takes in two inputs and gives two corresponding outputs. ....	32
Figure 14 Architecture for the Multi-Stage CVS Model. Every additional stage takes n new attention blocks. ....	34
Figure 15 Architecture for the RVS model which has 1 attention block for every data modality present in the RVS. ....	35
Figure 16 Attention mechanism for the RVS model. ....	36
Figure 17: Step by step implementation of stage-wise learning. ....	38
Figure 18: Example from the XMedia [11, 47] dataset. ....	40
Figure 19 Example from the XMediaNet [51, 59] dataset. ....	41
Figure 20: Sample images from the Nuswide dataset. [49] ....	42
Figure 21: A visualization of mean average precision. ....	44
Figure 22 The total loss for the CVS model as a function of the epochs visualized on Tensorboard. ....	48
Figure 23: The metric loss for the M-CVS model with respect to the total epochs. ....	50
Figure 24: The decrease in the cross entropy loss with respect to the total number of epochs in the RVS model. ....	51
Figure 25 Effect of all the 3 components of attention. ....	54
Figure 26: The total attention blocks that need to be trained when we add a new modality. ....	54

Figure 27: The time per iteration for the CVS and the M-CVS model as we increase the number of modalities added to the model. .... 55

Figure 28: Non converging of the reconstruction loss for the image and text modality..... 56



# List of Tables

---

Table 1: XMedia and XMediaNet dataset statistics. Figures indicate the number of training and testing samples.....	41
Table 2: XMedia and XMediaNet Features and their dimensions. ....	42
Table 3: Pascal and Nuswide dataset statistics. Figures indicate the train and test splits. ....	43
Table 4: Train and test split for the Birds dataset used to perform zero-shot retrieval. ....	43
Table 5: mAP scores for the Pascal and Nuswide dataset. ....	46
Table 6: mAP scores for the Xmedia and XMediaNet dataset. ....	47
Table 7: mAP scores for the images, text, and video for the XMediaNet dataset. ....	47
Table 8: Cross modal retrieval scores (mAP) for the five modalities in the XMedia dataset. ....	48
Table 9: Cross modal retrieval scores(mAP) for the M-CVS model on the XMedia datadset. ....	49
Table 10: mAP scores for images, text, and video for XMediaNet dataset. ....	50
Table 11: Cross modal retrieval mAP scores for the XMedia dataset. ....	51
Table 12: Stage wise learning methodology for the Pascal and Birds dataset. ....	52
Table 13: Stage-wise learning retrieval scores on Pascal and Birds dataset. ....	53

### 1.1 Introduction

Deep learning has shown great ability to perform targeted tasks like image classification, segmentation, object detection, and language translation. With the extensive availability of data and increased understanding of loss functions, neural networks are able to generate a good understanding of images, text, audio signals and videos. Convolutional Neural Networks (CNNs) have become the default model for computer vision tasks that were traditionally done using handcrafted features. CNNs architectures are now used for image classification, image segmentation, and object detection. Further, CNNs can be used as an image encoder, compressing a very high dimensional image into a small dimensional vector. Recurrent Neural Networks (RNNs) such as Long Short Term Memory (LSTMs) have proven to perform well in sequential tasks like language translation and caption generation. Within the past few years, Generative Adversarial Networks (GANs) have been shown to generate realistic-looking images from input noise vectors. The unique selling point of all these deep learning architectures is that they are able to learn the necessary parameters to accomplish real-world tasks. These deep architectures along with their learned parameters have been shown do a better job of most tasks when compared to the traditional machine learning methods paired with handcrafted features. Often, the deep architectures are able to develop a deeper understanding of the data which may not be easily perceptible to a human being.

These deep neural networks have shown incredible ability in targeted tasks in uni-modal settings like image classification or sentence to sentence translation. However, converting from one modality to another still remains a challenge. The work by Wu *et al.* [27] and Huang and Peng [26] show the ability to transfer from the image to text domain while [28] shows how we can transfer from audio to video domain. Vendrov *et al.* [24] and You *et al.* [25] show how images and text can be retrieved at the instance level whereas the works in [20, 21, 22, 23] show category level retrieval on images and text. Qi *et al.* [8] and Qi and Peng [10] showed cross modal retrieval using different learning techniques. Srivastava *et al.* [54] trained a Boltzmann machine that translates information from one modality to another.

More recently, adversarial loss functions and self-supervised learning techniques have been successfully shown to perform retrieval.

## 1.2 Motivation

Any data processed or stored by a computer ultimately needs to be converted into a binary format. These binary numbers are the most basic representations of data for computers. Similarly, vectors are the most basic structures of objects or concepts for machine learning models. Vectors are nothing but a list of numbers on which operations could be performed. These vectors will be used as the input and outputs to the deep neural networks used in this research.

Data originating from any modality (image, text, video, etc.) is first and foremost converted to a vector representation before being fed to a deep neural network. It is therefore very important that these vector representations of data are robust and are able to relay as much information about the input data as possible. Even if a model architecture is highly optimized and robust, unless the input data vectors are efficient, the model may end up failing.

This work aims to form efficient vector representations not only for a unimodal setting but also a multimodal setting where data belonging to different modalities can seamlessly interact with each other. The transfer of knowledge from one modality to the other is done through the medium of this common embedding space that we define.

We develop an architecture that is able to handle any number of modalities, and experiments will be conducted with up to five modalities. This work describes multiple ways in which a multimodal system can be built and provides a way of mapping new modalities into the model without having to change existing transformations and inferences. This work also compares different ways to formulate such a network and develops an understanding of the various pitfalls that these models face.

Such a model can be used for remote sensing applications where a vast range of sensors pick up data in different modalities to perform tasks like object detection and activity recognition. For example, a satellite can scan the ground and pick up passive signals like RGB images, infrared images, hyperspectral images as well as active signals like LIDAR point clouds, Synthetic-aperture radar (SAR) images etc. Similarly, sensors on the ground can pick up other information about the environment. Our model can then be used to help transfer contextual information from one modality to the other. Our model can also help in providing redundancy in

case of failure of any of the modalities. Similarly, such a model proves useful in mapping medical imagery data from sources like X-Rays, MRIs, CT Scans, ultrasounds etc.

The models we develop can be further used to train a network that takes in an input from one modality and generates synthetic data in a different modality. Such generation of data can prove to be helpful in under sampled modalities.

### **1.3 Contributions**

The main contributions of this thesis work can be summarized as:

- Extend the Common Vector Space (CVS) framework into two new architectures- the Multi-stage Common Vector Space (M-CVS) and the Reference Vector Space RVS and compare the three networks.
- Implement stage wise training of neural networks to see effect on performance of retrieval.
- Achieve robust transfer of information across multiple modalities using the same network architecture.
- Gain insight into the pros and cons of different training strategies.

### 2.1 Deep Learning

Deep learning is a branch of machine learning that deals with artificial neural networks which contain many weights or parameters, and often many hidden or intermediate layers. Deep learning works best when vast amounts of data and computing resources are available. It has provided massive breakthroughs in the fields of computer vision and language modeling. Deep learning architectures have a significant advantage over traditional methods that involve handcrafted features because the features are learned by the architecture. Additionally, the formulation of smarter loss functions has meant that deep learning can be used for a variety of tasks like image segmentation, object detection, and language translation.

### 2.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) form the backbone of modern computer vision processing and have helped advance the field of image processing. CNNs are able to effectively extract and learn features from gridded data structures like images or video frames. The evolution of CNNs from VGG-Net [2] to more complex architectures like the Inception [5] and ResNet [6] have helped achieve state-of-the-art results on multiple benchmark datasets like the CIFAR10 [7] and ImageNet [17]. CNNs perform very well in image classification [2, 18, 6, 19], image segmentation [4], and object detection [3].

CNNs generally consist of three basic operations: convolution, pooling, and fully connected layers. The convolution layers use filter kernels to perform convolution operations on the input before passing it on to the next layer. The pooling layers down sample the input that it receives whereas the fully connected layers form multi-layer perceptron layers to finally arrive at the sample classification. At a high level, the convolution and pooling layers extract a salient hierarchy of features, and the fully connected layers perform the classification. One reason this architecture is so powerful is that both the features and classifier components are learned simultaneously through a process called backpropagation.

The convolution operation typically preserves the output dimensions of the input provided necessary padding of the original input is done. On the other hand, pooling decreases the dimensions of the data. Average pooling and max-pooling are the two most popularly used pooling techniques. Figure 1 shows a typical CNN architecture for an image classification setting. An input image passes through various convolution and pooling layers. Finally, it is flattened out to form a single vector. This single vector is then passed through multiple fully connected layers after which a final layer performs classification. The connections between layers use non-linear activation functions like the sigmoid, tanh and relu so that the network can learn some very complex patterns.

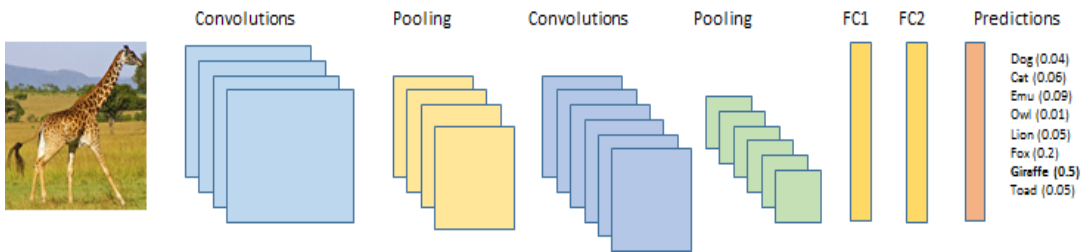


Figure 1: Typical CNN architecture. An image passes through multiple convolutions and filtering operations before being passed through fully connected layers and classified into a one out of several classes.

Equation (2.2.1) explains the convolution operation.  $I^l_{(x_r, x_h, c)}$  is the image at the  $x_r^{th}$  and  $x_h^{th}$  pixel at the  $c^{th}$  channel for the  $l^{th}$  layer of the network. The image  $I$  is convolved with a filter kernel  $K$  to get the output of the convolution operation  $M$ .

$$M^l_{(i, j, k)} = K_{abc} I^{l-1}_{(x_r+a, x_h+b, c)} \quad (2.2.1)$$

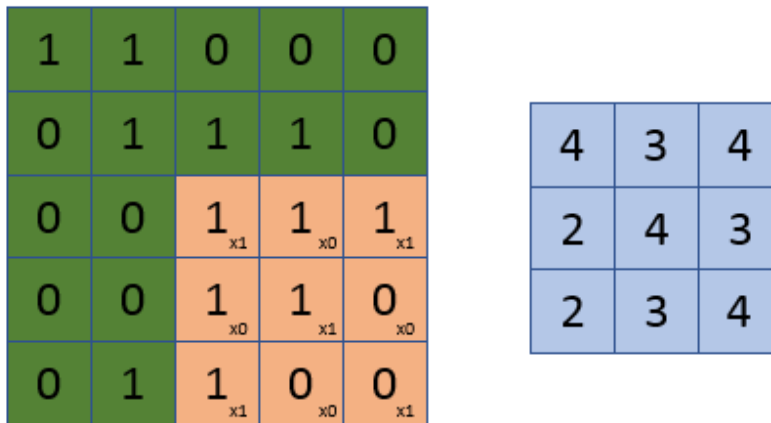


Figure 2: Convolution Operation. Left – Input array; right – convolution kernel.

In Figure 2, the input array (image) is convolved with the filter to form a new output array. This filter is a matrix of parameters that are learned during backpropagation in a CNN. The learning of many such filters along with the fully connected layers enable the CNN to learn abstract concepts.

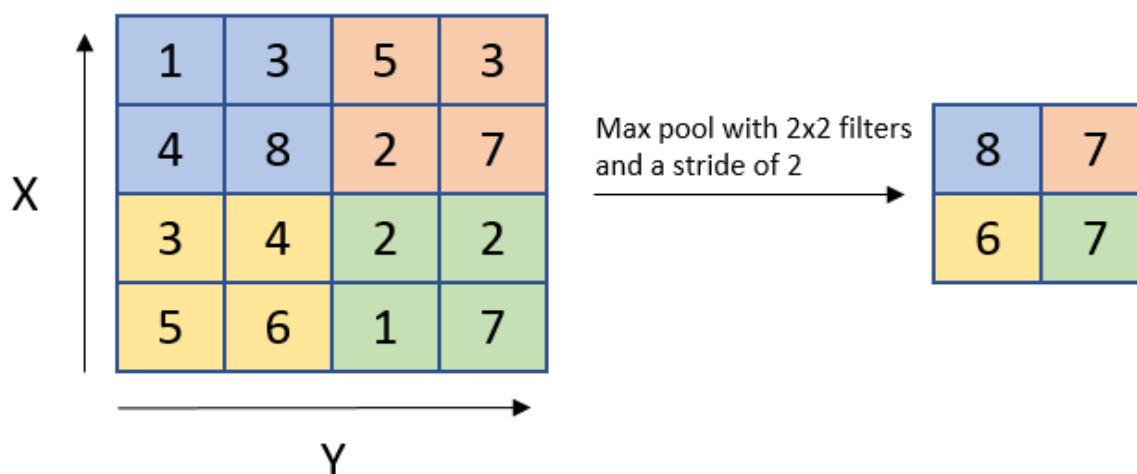


Figure 3: Max pooling operation in a convolutional neural network.

Figure 3 shows a max-pooling operation where the maximum value of the cells belonging to the same color is passed to the next layer.

Videos are essentially a sequence of images and hence, CNNs are useful for encoding videos too. Karpathy *et al.* [31] showed an efficient way to encode videos into their vector representations. Recent works like [32, 33] show different ways in which videos can be encoded into a vector representation for tasks such as classification or segmentation.

## 2.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) along with Long Short Term Memory (LSTMs) units and Gated Recurrent Units (GRUs) help encode and decode sequential data. These RNNs have the ability to retain important information in the sequence. They are used extensively to represent textual data. RNNs are used in tasks like image captioning and sentence encoding.

Work in natural language processing has led to the development of architectures that are able to perform tasks like image captioning [14 15] and video summarization [16]. Kiros *et al.* [13] showed how sentences can be encoded into vectors.

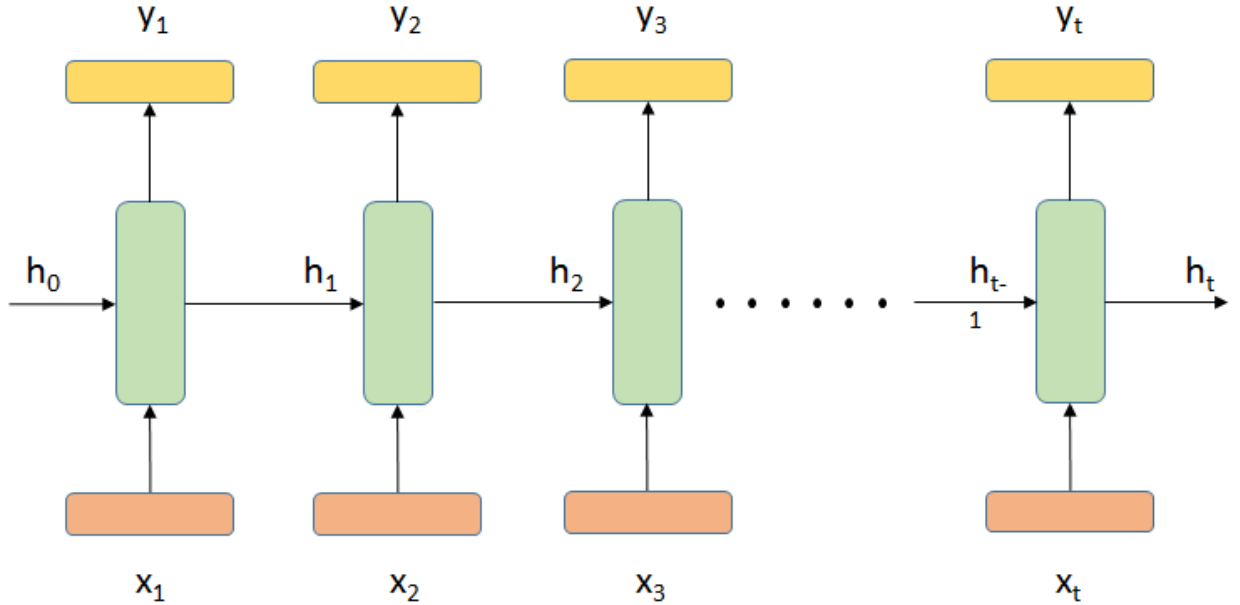


Figure 4: A recurrent neural network consisting of many sequential cells that take input from the previous state and also the current input to predict the output at each time step.

In the Figure 4,  $x_1, x_2 \dots x_t$  are the inputs for time step  $t = 1$  to  $t$  and  $y_1, y_2 \dots y_t$  is the predicted next outputs.

$$h_t = f(W^{hh}h_{t-1} + W^{hx}x_t) \quad (2.3.1)$$

$$y_t = \text{softmax}(W^sh_t) \quad (2.3.2)$$

$$J^{(t)}(\theta) = \sum(y_{ti} \log y_{ti}) \quad (2.3.3)$$

The above equations describe the working of the RNNs. The information about the previous time steps is held in (2.3.1) where  $h_t$  is calculated based on  $h_{t-1}$ .  $h_0$  is set to a vector of zeros. A sigmoid activation is then applied to the final summation as seen in (2.3.2). In (2.3.3), the loss is calculated at each time step to find out the error between the input word and the predicted word.



RNNs face the problems of vanishing gradients and hence LSTMs and GRUs have become the default way to model sequential data.

### A. Long Short Term Memory (LSTM)

The vanishing gradient problem with RNNs is overcome using LSTMs and GRUs. LSTMs are able to selectively remember both long and short-term information. They were introduced in [34] which outperformed vanilla RNNs.

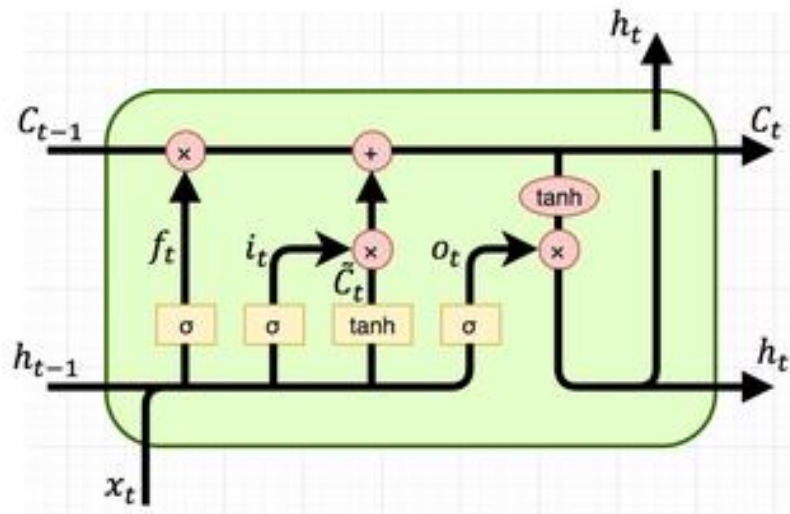


Figure 5: LSTM cell with input, output and forget cells that allows it to retain important information. [62]

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \quad (2.3.4)$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \quad (2.3.5)$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \quad (2.3.6)$$

$$\hat{C}_t = \tanh(x_t U^g + h_{t-1} W^g) \quad (2.3.7)$$

$$C_t = \sigma(f_t * C_{t-1} + i_t * \hat{C}_t) \quad (2.3.8)$$

$$h_t = \tanh(C_t) * o_t \quad (2.3.9)$$

Equations (2.3.3) – (2.3.9) describe the mathematics of the LSTM cell. Figure 5 shows the internal diagram of the LSTM cell. Here,  $I, f, o$  shown indicate input, forget and output gates respectively.  $W$  indicates the recurrent connection from the previous hidden state and the  $U$  is the weight matrix that transforms the inputs to the current hidden layer. The sigmoid function converts all the values to (0, 1) and helps define the information that can flow through the gates.  $\hat{C}_t$  is the hidden state while  $C$  is the internal memory unit. This memory unit helps to combine the current inputs and context from the previous hidden states to generate a better vector representation of the input. It is this vector representation that we generally use to encode our text data.

### B. Gated Recurrent United (GRU)

GRUs is a variant of LSTMs which have fewer parameters and easier to train on. It was introduced by Cho *et al.* [35]. Figure 6 shows the structure of a GRU.

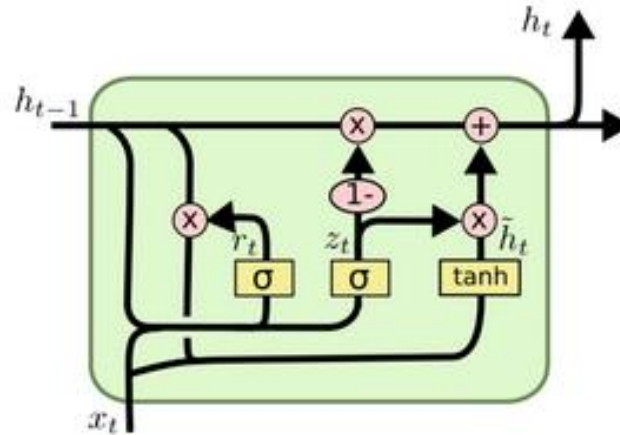


Figure 6: A GRU unit which has comparatively fewer parameters and is easier to train and tune. [62]

$$z_t = \sigma(x_t U^z + h_{t-1} W^z) \quad (2.3.10)$$

$$r_t = \sigma(x_t U^r + h_{t-1} W^r) \quad (2.3.11)$$

$$\hat{h}_t = \tanh(x_t U^h + (r_t * h_{t-1}) W^h) \quad (2.3.12)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \quad (2.3.13)$$

Equations (2.3.10) – (2.3.13) describe the mathematical operation of the GRUs. The  $r$  is the reset gate while  $z$  is the update gate. The reset gate measures how much of the new input will be combined with the memory and the update gate defines how much of the previous memory will be kept. Assigning reset to 1 and update to 0, we have the original RNN model.

## 2.4 Multi-modal and Cross-modal Retrieval

Cross-modal retrieval has gained significant traction in the last decade due to the availability of vast amounts of data. The information retrieval from one modality to another is a challenging task because of the difference in their statistical properties. Most traditional techniques involve the formulation of a latent space in which the properties of the two modalities can be matched. Canonical Correlation Analysis (CCA) [53] is one technique that learns to maximize the correlation of the data belonging to different modalities.

More recently, with the development of deep learning techniques, the field of cross-modal retrieval has seen tremendous development. Deep learning has helped improve the way in which data can be encoded into vectors. Similarly, models have been built that have transferred knowledge across multiple modalities. Most work in the multi-modal space restricts itself to only two modalities. There are two types of works in the cross-modal retrieval space: 1) cross-modal hashing [36, 37] which maps the data into a common space in a computationally efficient manner which would make retrieval faster; and 2) real-value based cross-modal retrieval which uses different loss functions to perform retrieval.

Zhang *et al.* [60] developed a framework that generated vector representations of sentences while Sah *et al.* [61] used multiple captions to generate images from common vector representations. Deng *et al.* [45] performed cross-modal hashing using a triplet based hashing network and Wu *et al.* [46] showed the use of adversarial training by using cycle consistency loss function. The task of real-value based retrieval can be modeled in different ways. Qi *et al.* [8] used the triplet loss function to form triplets (anchors, positive, negative) samples and train

a model whose objective function ensured that the encoded vector representation of anchors and their corresponding positive samples lie close in the latent space, while the encoded vector representation anchors and negative samples are pushed far away. Feng *et al.* [22] used correspondence auto encoders while Qi and Peng [9] used reinforcement learning to perform cross-modal retrieval.

Zhen *et al.* [43] used a combination of a modality invariance loss, linear classifier, intermodal and intramodal discriminative loss to perform retrieval as shown in (2.4.1) and (2.4.2) and (2.4.6) respectively. Figure 7 shows the architecture used by the DSCMR [43] model. The image and text are passed through separate CNNs and their feature vectors are extracted. The losses described in (2.4.1), (2.4.2) and (2.4.6) are then calculated on these common space vectors.

$$L_{inv} = \frac{1}{N} \|U - V\|_F \quad (2.4.1)$$

$$L_{class} = \frac{1}{N} \|P^T U - Y\|_F + \frac{1}{N} \|P^T V - Y\|_F \quad (2.4.2)$$

$$L_{inter} = \frac{1}{N^2} \sum_{i,j=1}^N (\log(1 + e^{\hat{\Gamma}_{ij}}) - S_{ij}^{\alpha\beta} \hat{\Gamma}_{ij}) \quad (2.4.3)$$

$$L_{img} = \frac{1}{N^2} \sum_{i,j=1}^N (\log(1 + e^{\Phi_{ij}}) - S_{ij}^{\alpha\alpha} \Phi_{ij}) \quad (2.4.4)$$

$$L_{txt} = \frac{1}{N^2} \sum_{i,j=1}^N (\log(1 + e^{\theta_{ij}}) - S_{ij}^{\beta\beta} \theta_{ij}) \quad (2.4.5)$$

$$L_{modal} = L_{inter} + L_{img} + L_{txt} \quad (2.4.6)$$

Where:

- $\hat{\Gamma}_{ij}$  is the cosine distance between two data points belonging to different modalities.

- $\Phi_{ij}$  is the cosine distance between two data points belonging to image modality.
- $\theta_{ij}$  is the cosine distance between two data points belonging to text modality.
- $S_{ij}^{\alpha\alpha}, S_{ij}^{\alpha\beta}, S_{ij}^{\beta\beta}$  are indicator functions whose value is 1 if the two elements are of same class and 0 otherwise.

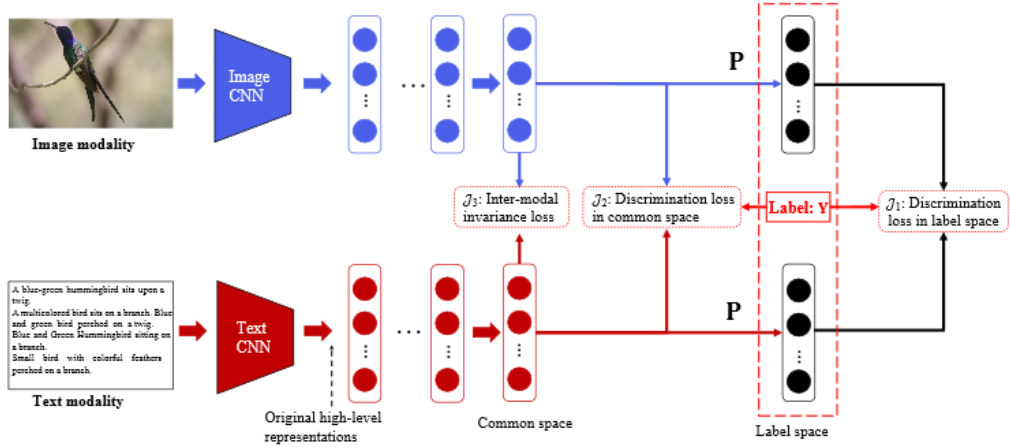


Figure 7: The DSCMR [43] architecture consisting of three separate loss functions to perform cross modal retrieval.

Xu *et al.* [42] used adversarial loss functions and built an architecture that converted a 4096-dimensional vector representation to a 200 dimension vector using fully connected layers. Peng *et al.* [47] created an architecture that was able to retrieve images from their outline sketches and vice-versa. Works in [55, 56] show the use of retrieval of data from one modality to the other at the granular level where the input query is able to retrieve its corresponding ground-truth sample of the other modality.

Many image captioning works like [39, 40] use attention mechanism to better align words and image regions. Similarly, the attention mechanism proposed by Lee *et al.* [41] can be used for a better correlation between vectors belonging to different modalities. Attention models provide important context to different parts of the input text and the different regions of the image. This idea can be extended to other modalities as well if attention is applied to multi-dimensional feature vector that is formed after the encoding network.

In our work, we focus on the CUB-Birds [58], Pascal [50], NUSWIDE [49] datasets have data belonging to two modalities and also extend the ability of our model to handle up to five modalities where we test our model on the XMedia [47, 48] and XMediaNet [51, 59] datasets.

## 2.5 Metric Learning Loss Functions

The backbone of all deep learning applications is to define an appropriate loss function for the required task. Metric learning is a type of learning that is able to map similar representations of data between two or more modalities. Metric learning involves the process of formulating pairs of positive and negative data to achieve the required alignment in the data. The following are some of the popular metric learning loss functions that have been developed.

### A. *Contrastive Loss Function [38]*

Introduced by Hadsell *et al.* [38], the contrastive loss function calculates the distance between the encoding of the image and the text data points. The goal of the loss function is to have the negative pairs at least a distance of *margin* away from the positive pairs where the distance calculated is the Euclidian distance between two points.

$$L_c = \frac{1}{2N} \sum ((y)d^2 + (1 - y) \max(\text{margin} - d, 0)^2) \quad (2.5.1)$$

Where

- $d$  is the distance between image and sentence pair
- $y$  is 1 if the two samples are similar and 0 if they are dissimilar.

### B. *Triplet Loss Function [12]*

The triplet loss function forms triplets of data instead of pairs that are formed by the contrastive loss. Schroff *et al.* [12] developed this loss where an anchor, positive and a negative sample is selected. The objective of the loss function is to minimize the distance between the anchor and the positive sample while keeping the distance between the anchor and negative sample at least a margin apart, where the margin is similar to that used in the contrastive loss. The distance function used in the triplet loss is the Euclidian distance. Figure 8 depicts the different circumstances when the triplet loss will be updated.

$$L_T = \frac{1}{2N} \sum \max(0, |f_a^i - f_p^i|^2 - |f_a^i - f_n^i|^2 + \text{margin}) \quad (2.5.2)$$

Where

- $f_a^i$  is the feature embedding of the anchor
- $f_p^i$  is the feature embedding of the positive sample
- $f_n^i$  is the feature embedding of the negative sample

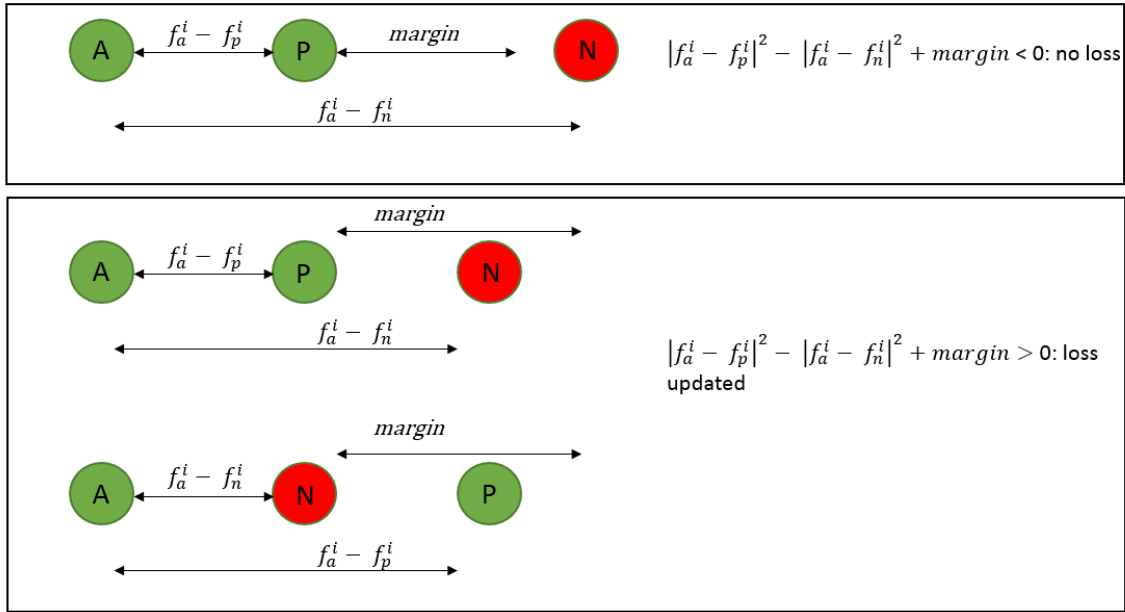


Figure 8 Cases for the triplet loss. Loss is only updated in the bottom two cases when the negative is closer to the positive and the anchor.

### C. Adversarial Loss Function [57]

Adversarial loss functions have been successfully able to generate images from noise vectors using discriminator and generator networks. There are two goals of adversarial training: 1) to train a discriminator network that is unable to distinguish between real and fake data, and 2) train a generator network that can seamlessly generate fake samples of data that will be able to fool the discriminator.

$$(\min G, \max D) L(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{x \sim p_z(z)} [\log (1 - D(G(z)))] \quad (2.5.3)$$

Where

- $D$  – Discriminator whose goal is to be able to distinguish between real and fake data

- $G$  – Generator whose goal is to build a network that is very good at fooling the discriminator
- $x, z$  – real and generated samples

## 2.6 Semi-Hard Negative Mining

In practice, it is easy to find pairs of positive and negative samples that are a margin distance apart. Since these pairs already satisfy the underlying condition, they will not contribute to the loss calculation. Semi-hard negative mining used by Schroff [12] *et al.* is a technique where the harder of the negative samples are used to calculate the loss and allow for better and faster convergence of the network.

The negative samples can be categorized as below:

- hard – they are closer to the anchor than positive samples
- Semi-hard – they lie within margin distance from the positive samples
- easy negatives – they lie beyond the positive samples

The selection of easy negatives results in a loss value of zero whereas the selection of hard negatives results in a loss value that could be too high for the model to handle. Therefore, the semi-hard negatives are just the right negative samples that can help our model converge.



In this section, we introduce the various architectures and networks that will be used in this research and explore the differences between them. Figure 8 describes an ideal transformation that will happen with a common vector representation. Initially, when the model is randomly initialized, we see that all the points in the space are scattered without any clusters being formed. The right half of Figure 9 depicts how the space looks after training the model and minimizing the loss values for the entire model. The different colors in the figure represent the various modalities of data like images, text, audio, video, etc. whereas the shapes represent the categories (cars, boats, people, tigers, etc) associated with these data points.

The training process yields a transformation function for any given data point into the vector space. A test sample will undergo the same transformation and will be projected into the space. Ideally, the test sample should lie very close to other samples of the same category.

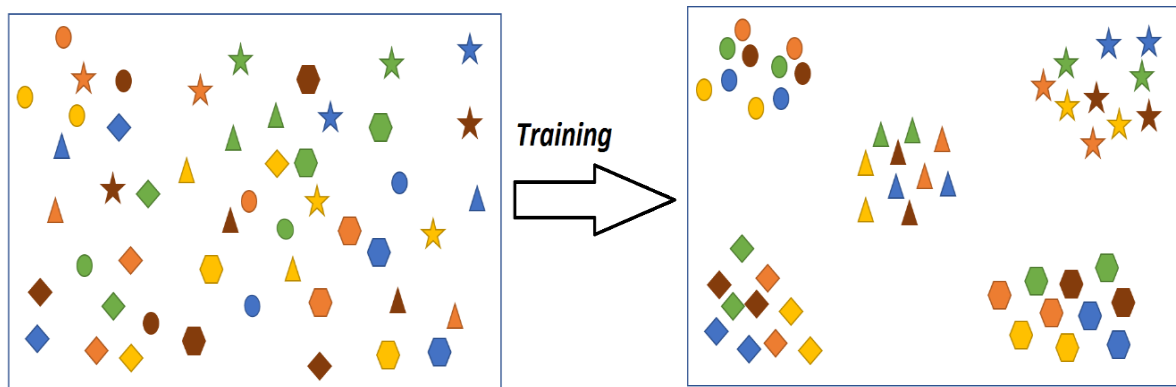


Figure 9: The vector space before (left) and after (right) training. The colors represent the different data modalities and the shapes represent the categories.

### 3.1 Common Vector Space (CVS)

The common vector space architecture (CVS) takes in data belonging to different modalities and dimensions and maps them on to a lower-dimensional common embedding space. The CVS model is developed for two modalities and extended up to five modalities. The CVS uses different encoders to extract feature vectors and uses fully connected neural networks to map into the common space. Figure 8 shows the architecture overview of the CVS model with five input modalities. CNNs are used as the default encoding network for images and videos, audios are encoded using MFCC features whereas text data can be encoded using a bag of words model or a sentence-to-vector representation such as SkipThoughts [13].

These encoders convert a higher dimensional data from fully connected layers, into a vector representation to further reduce the dimensions of these vectors. To simplify, (3.1.1) shows the generic transformation of the input into the CVS representation.  $X$  is the input data which could either be images, text, video, audio or 3D,  $E$  is the common vector space representation,  $W^E$  is the encoder weight matrix, and  $W^T$  is the weight matrix that transforms the feature vector into the common embedding vector representation.

$$E = W^T (W^E X) \quad (3.1.1)$$

Figure 10 depicts the architectural set up of the CVS model. In Figure 9, the vector representations are indicated by  $h_i$ ,  $h_v$ ,  $h_a$ ,  $h_v$  and  $h_{3d}$ . Each modality has its separate encoder that converts the raw data into a vector representation. These vector representations are of different dimensions. For example,  $h_i$  is a 4k vector from ResNet [6], while  $h_a$  is a 29 dimensional MFCC feature, and so on. To bring all modalities to the same dimensions, we add two fully connected neural network layers, each containing 512 neurons. These layers along with the attention layer are initialized randomly and trained to minimize the total loss of the system.

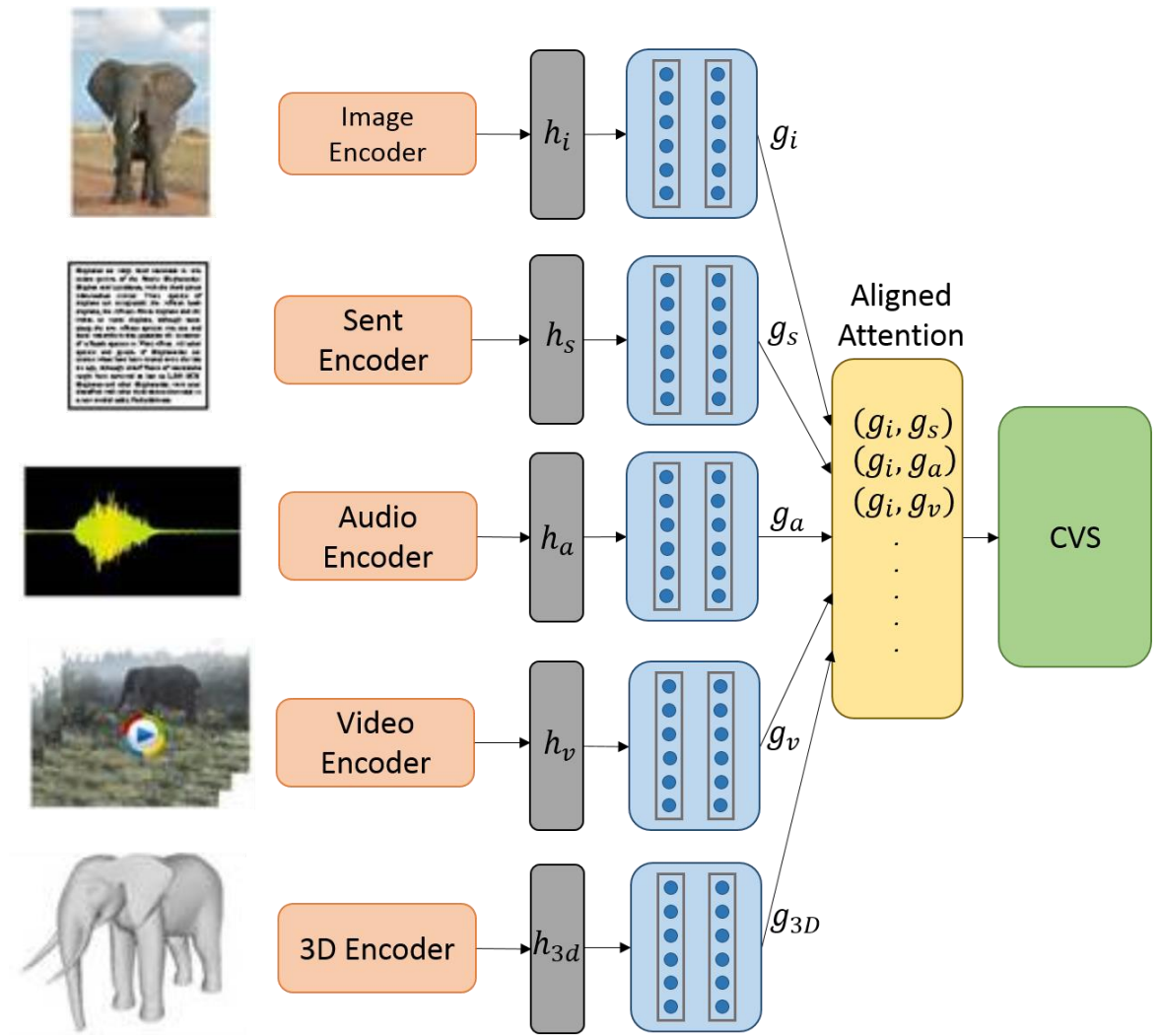


Figure 10 Architecture for the CVS model. Data belonging to different modalities will get encoded into vectors after which they are projected into the common embedding space using fully connected layers.

Once the input data is projected onto the CVS, we use the cross-entropy loss function to perform classification and use metric learning losses simultaneously to cluster the samples. In this embedding space, the points are rearranged such that the total entropy of the system is minimized. In the CVS, objects that belong to the same category are mapped close to each other and well separated from objects that belong to some other category irrespective of their modality.

For example, the images, text, and videos of an elephant will be well separated from the images, audio, and video of a plane. The objective of the training is to minimize the inter-cluster distance and maximize the intra-cluster distance. The distance between any two vectors can be calculated as the Euclidian distance as shown in (3.1.2) or the cosine distance between

two vectors as shown in (3.1.3). Figure 11 gives us an intuitive idea of how cosine distance works. The cosine distance gives importance to the angular rotation between two vectors and gives a similarity score.

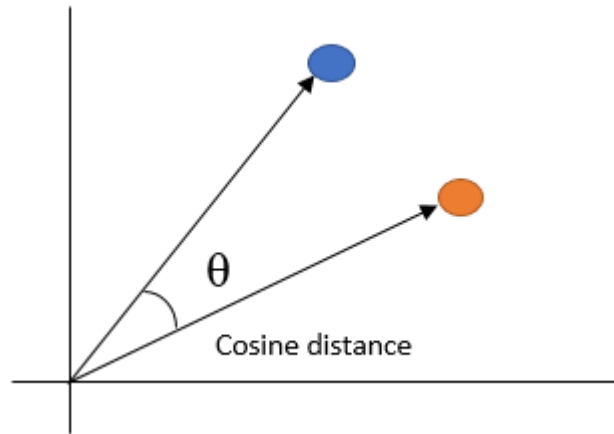


Figure 11 Cosine distance between two vectors in a 2D representation.

$$d = \|f_i - f_j\|_2^2 \quad (3.1.2)$$

$$d = \frac{\cos(f_i, f_j)}{\|f_i\| \cdot \|f_j\|} \quad (3.1.3)$$

### A. Loss Functions

The CVS uses three loss functions to minimize the entropy of the common space. These loss functions are common across the variations of the model that we describe in Sections 3.2 and 3.3. Figure 12 shows the adversarial loss function in this context of this model. The adversarial loss function has two separate loss values that are back propagated. The image reconstruction loss tries to minimize the difference between the input image embedding and the output image embedding while the sentence reconstruction loss minimizes the difference in the sentence embeddings.

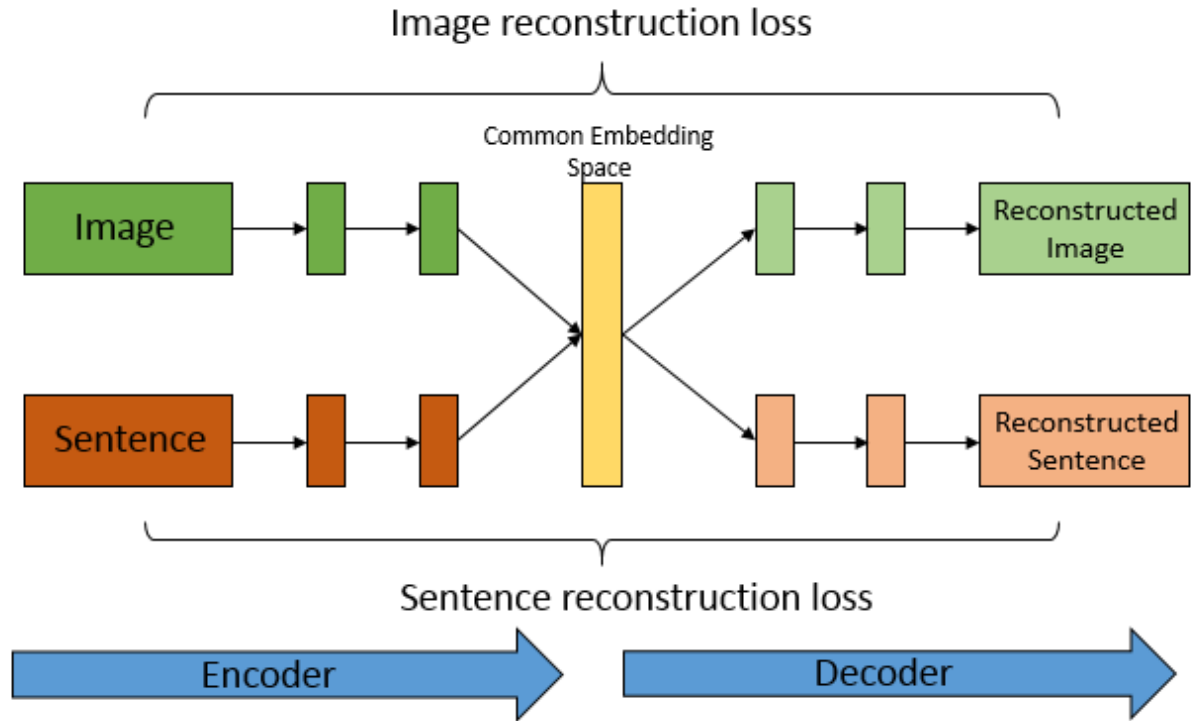


Figure 12: The adversarial loss function is calculated by comparing the generated and input images and sentences.

Equation (3.1.4) describes the classification loss for the CVS model where

- $n$  is the total number of the samples,
- $y$  is the ground truth
- $f(s)$  is the sigmoid function as shown (3.1.5).

$$L_c = -\frac{1}{n} \sum_{i=1}^n (y_i \log f_i(s) + (1 - y_i) (1 - \log f_i(s))) \quad (3.1.4)$$

$$f(s) = \frac{e^{s_i}}{\sum_j^c e^{s_j}} \quad (3.1.5)$$

A modified version of the triplet loss described in Section 2.5 is used in (3.1.6). The triplet loss we implement has two parts: inter-modal and intra-modal triplet loss. The loss is calculated for two modalities at a time. The inter-modal loss consists of the triplets between the two modalities. The intra-modal loss will calculate the triplet loss for the data points belonging to the same modality. We use a weighted sum of these losses to calculate the total triplet loss.

$$L_t = \sum_{i,s,v,a,3d} \gamma_1 \sum_{x,y} \max(0, |f_a^x - f_p^x|^2 - |f_a^x - f_n^x|^2 + \alpha_1) + \gamma_2 \sum_{x,y} \max(0, |f_a^y - f_p^y|^2 - |f_a^y - f_n^y|^2 + \alpha_2) + \gamma_3 \sum_{x,y} \max(0, |f_a^x - f_p^y|^2 - |f_a^x - f_n^y|^2 + \alpha_3) \quad (3.1.6)$$

Where:

- $|f_a^x - f_p^x|^2$  is the distance between the anchor and positive sample.
- $|f_a^x - f_n^x|^2$  is the distance between the anchor and negative sample.
- $\alpha_1, \alpha_2, \alpha_3$  are the margins for the triplet loss.
- $\gamma_1, \gamma_2, \gamma_3$ : are the individual weights of the 3 losses.

A weighted combination of  $L_c$  and  $L_t$  and are used in (3.1.8) to calculate the total loss of the system where  $\alpha$  and  $\beta$  are individual weights given to these two loss terms. We observe that the adversarial loss does not improve the model much. In fact, when used individually, it is unable to perform at the required level. We further discuss this phenomenon in Chapter 5.

$$L = \alpha L_c + \beta L_t \quad (3.1.7)$$

## ***B. Training Strategy***

Similar to most deep learning networks, the CVS is trained in batches. In the implementation of the triplet loss, for any iteration, an anchor, positive and negative sample is selected. Since the CVS model can handle up to five modalities, this selection happens in batches continuously till we have traversed through the entire dataset. In the CVS setting, all the information for all

the modalities is given to the model at the initial stage. If new data is to be added to the network, the entire network has to be re-trained.

In the CVS setting, we will always need  $\binom{n}{2}$  attention blocks where  $n$  is the total number of modalities. In the case of  $n = 5$ , we need 10 attention blocks.

Such a training methodology means that the CVS architecture can only be built once we have all the data that is going to be used.

### C. Aligned Attention

Attention has proven to be an effective mechanism in aligning two vectors. The alignment provides a way to map local features across the two modalities. Attention has shown to be effective in tasks such as image captioning [39, 40]. For a given duplet of vectors, attention calculates a similarity score  $e^t(i)$  for every feature in the vector where  $i$  is the  $i^{\text{th}}$  feature and  $t$  is time step. Computation of  $e^t_i$  is shown in (3.1.7)

$$e_i^t = W^T(W_a h_{t-1} + U_a v_i + b_a) \quad (3.1.8)$$

Where

- $h_{t-1}$  is the previous hidden state.
- $V_i$  is the  $i^{\text{th}}$  feature.
- $W_a, W_b, U_a, b_a$  are all learnable parameters.

$e^t_i$  is then passed through a softmax layer as (3.1.8). The final feature is a weighted combination of all the N input features as shown in (3.1.9).

$$a_i^t = \frac{\exp(e_i^t)}{\sum_{k=1}^N \exp(e_k^t)} \quad (3.1.9)$$

$$\theta_t(V) = \sum_{k=1}^N a_k^t v_k \quad (3.1.10)$$

$\theta_t(V)$  is used as the final output of the attention layer which transforms the input vector into a new aligned vector that is better aligned with each other.

In the multi-modal CVS setting, the output of the encoder network is fed to the attention model. The attention model as shown in Figure 13 takes in two input embeddings and transforms into an attended embedding. These embeddings are  $L_2$  normalized at the output stage before being projected into the CVS space. For any given two modalities, there has to exist one attention block between them. This pairwise attention block ensures that a pair of data points that belong to the same class are well aligned. Equation (3.1.10) shows how two input embeddings are converted to a single value at any given time step.

- $g_i$  – input embedding of modality 1.
- $g_s$  – input embedding of modality 2.
- $g_{i_s}$  – the element-wise product of  $g_i$  and  $g_s$ .
- $W_i, W_s, W_e$  – learned weights.

Equation (3.1.12) passes  $e_{i_s}^t$  through a softmax layer.  $e_{i_s}^t$  is the same dimension as the CVS dimension.  $\sum \exp(e_{i_s}^t)$  is a scalar quantity that is the sum of all the elements in the  $e_{i_s}^t$  vector. Thus, equation (3.1.12) provides the softmax functionality. In the attention mechanism, we insert a ResNet like skip connection so that the weight matrices just have to learn the difference between the input and the output and not the entire transformation itself. This reduces learning time and improves results in our experiments and is a key component of the aligned attention block.

$$e_{i_s}^t = W_e \tanh(g_i W_i + g_s W_s + g_i \cdot g_s W_{i_s}) \quad (3.1.11)$$

$$\alpha_{i_s}^t = \frac{\exp(e_{i_s}^t)}{\sum \exp(e_{i_s}^t)} \quad (3.1.12)$$

Equation (3.1.13) and (3.1.14) shows the output of the attention block. The  $\alpha_{i_s}^t$  is multiplied with the input and added to the output.

$$g_i = g_i + \alpha_{i_s}^t \cdot g_i \quad (3.1.13)$$



$$g_s = g_s + \alpha_{is}^t \cdot g_s \quad (3.1.14)$$

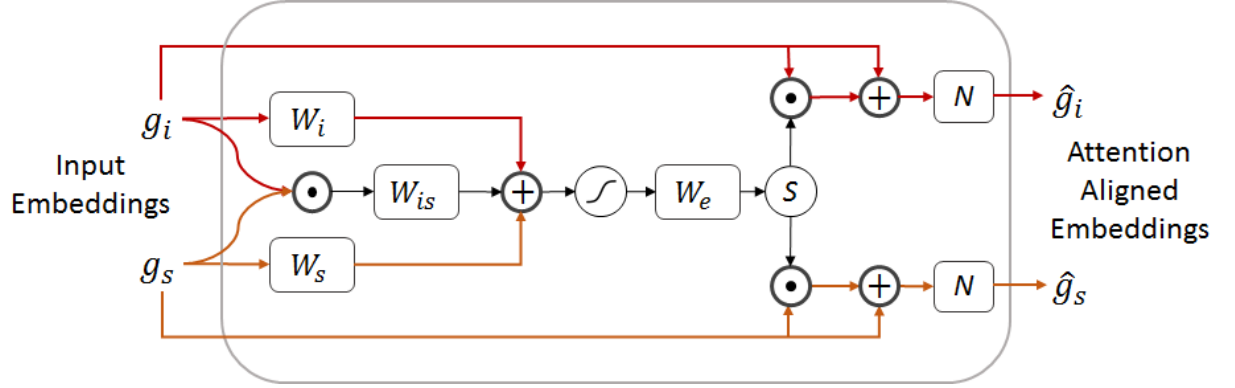


Figure 13 Aligned attention block which takes in two inputs and gives two corresponding outputs.

#### D. Limitations of the CVS Model

The CVS model has some limitations:

- Every time a new modality is added to the CVS, the entire architecture will be reset and we need to restart training from scratch.
- There is no provision in the structure of the CVS to perform incremental stage-wise training.
- Since the model will re-train, we lose traction of the existing inferences. This makes the CVS model inflexible to the addition of new modalities.

We try to address some of these issues by introducing Multi-staged Common Vector Space (M-CVS) and Reference Vector Space (RVS) respectively, in Sections 3.2 and 3.3.

### 3.2 Multi-Staged Common Vector Space (M-CVS)

The Multi-Staged Common Vector Space (M-CVS) is a space that is the same as the CVS except the training procedure is different. The objective here is still the same as the CVS and similar objects and concepts will lie close to each other in this M-CVS. To overcome the shortcomings of the CVS model, we develop a training strategy that builds the CVS architecture in a stage-wise manner. The M-CVS model network is able to add newer modalities of data in a stage-wise manner without having to retrain or change the existing network architecture. This means that the M-CVS model can be initially trained on two

modalities and at a later time, an additional third, fourth and fifth modality can be added without having to change the original two modality network.

In the M-CVS model, the updating of weights in the back-propagation is controlled such that only weight matrices belonging to the new modality of data are updated whereas the other weights remain unchanged. For such an architectural setup,  $n$  new attention blocks are added in every stage.

Figure 14 shows the architecture of the multi-stage CVS model. The data belonging to different modalities passes through separate encoding networks and fully connected layers before being passed into the M-CVS layer. In every subsequent stage, newer modalities along with the new attention blocks are added. For example, in stage 2, the video modality is added to the M-CVS model and an additional two attention blocks are introduced. The attention block used in the M-CVS model is the same as described in Chapter 3.

### ***A. Training Strategy***

The M-CVS model uses the same feature extractors as the CVS model. The advantage of defining the M-CVS is that it allows the addition of new information to the vector space without disturbing existing information in the space. For instance, if the image and text modalities were present in the M-CVS, the addition of the video modality will not affect the existing transformation between the images and text modalities.

# Architecture : Multi Stage CVS

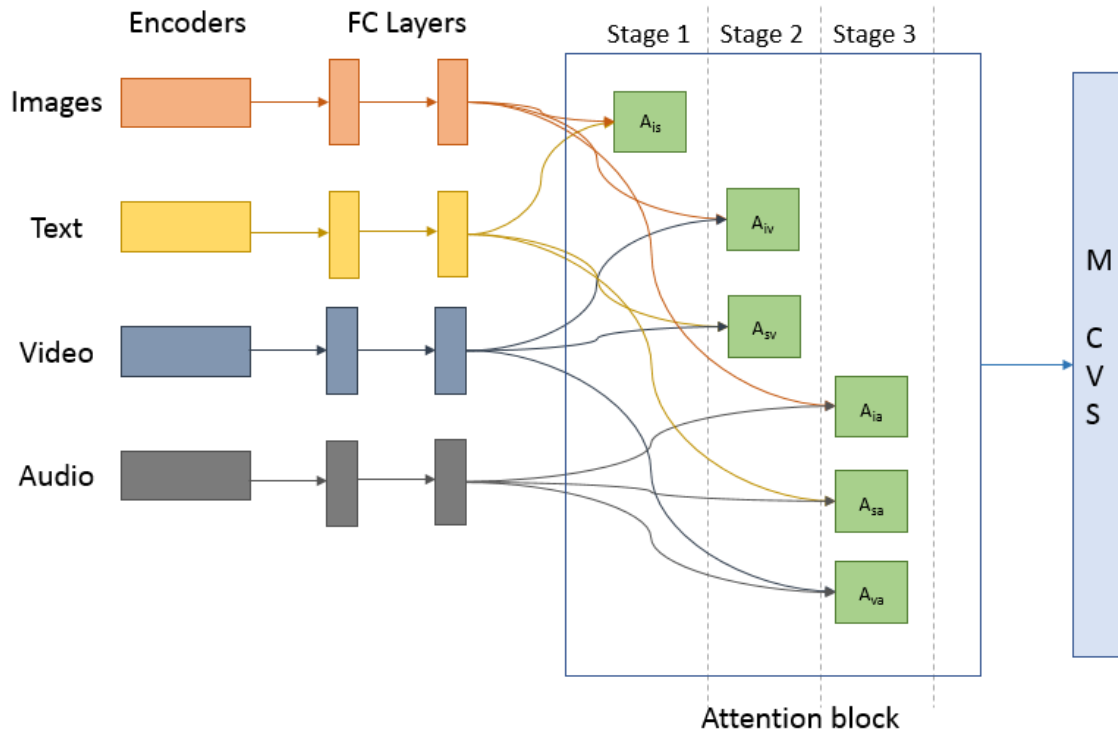


Figure 14 Architecture for the Multi-Stage CVS Model. Every additional stage takes  $n$  new attention blocks.

## 3.3 Reference Vector Space (RVS)

The Reference Vector Space (RVS) is again, conceptually the same as the CVS and the M-CVS. The RVS goes one step further in restricting the number of attention blocks required at each stage. The main idea of the RVS is to be able to have one transformation for every modality into the common embedding space. The RVS is inspired by the International Color Consortium's device independent profile connection space where inputs from multiple sources are mapped to a common reference color specification that allows easy transfer from one color space to any other.

In the RVS, we define one reference modality to which all other modalities will be mapped to. This allows the mapping of all new modalities through this singular reference modality. Similar to the CVS and the M-CVS architecture, the goal is to have similar objects and concepts lie close to each other in clusters that are well separated from groups of other concepts and objects in the RVS. Figure 15 shows the architecture set up of the RVS model where only one additional attention block is required to map the new modality into the RVS.

### A. Aligned Attention for RVS

The attention mechanism developed for the RVS model is such that there are only  $n$  attention blocks for  $n$  modalities. In the CVS model, there were  $\binom{n}{2}$  attention blocks whereas, in the M-CVS model, there are  $n$  additional blocks for a newly added modality. For the RVS model, there is only 1 additional attention block added per new modality. Figure 16 describes the modified attention mechanism for the RVS. Equation 3.3.1 describes the calculations for the aligned attention for the RVS.

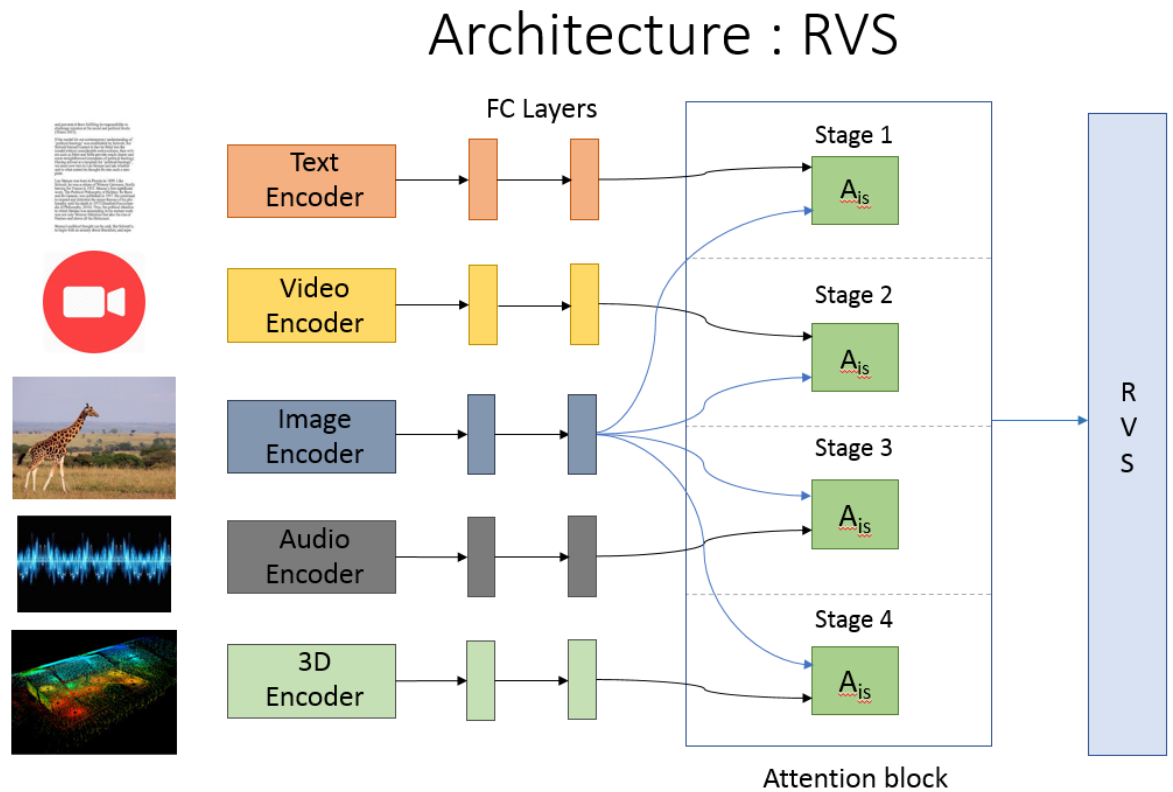


Figure 15 Architecture for the RVS model which has 1 attention block for every data modality present in the RVS.

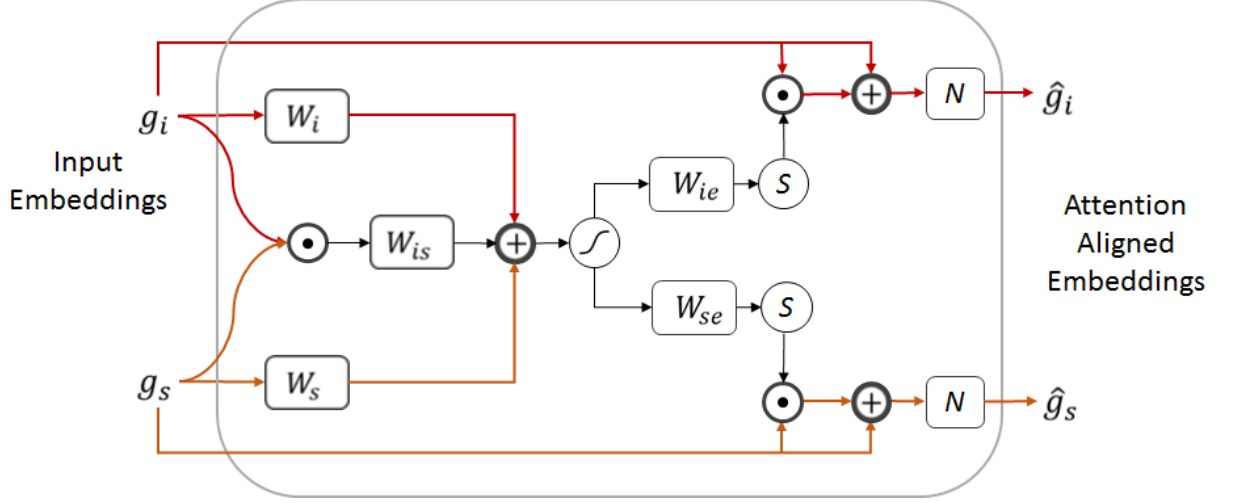


Figure 16 Attention mechanism for the RVS model.

We use a similar attention model as described in Chapter 3 but modify it to satisfy the needs of the RVS. We break down the  $W_e$  from 3.1.11 into two parts  $W_{ie}$  and  $W_{se}$  where  $W_{ie}$  is the reference weight that does not change after the initial training. However,  $W_{se}$  changes after the addition of every modality. So, for each additional modality that is added to the architecture, we will add a new  $W_{se}$ .

$$e_{ie}^t = W_{ie} \tanh(g_i W_i + g_s W_s + g_i \cdot g_s W_{is}) \quad (3.3.1)$$

$$e_{se}^t = W_{se} \tanh(g_i W_i + g_s W_s + g_i \cdot g_s W_{is}) \quad (3.3.2)$$

$$\alpha_{ie}^t = \frac{\exp(e_{ie}^t)}{\sum \exp(e_{ie}^t)} \quad (3.3.3)$$

$$\alpha_{se}^t = \frac{\exp(e_{se}^t)}{\sum \exp(e_{se}^t)} \quad (3.3.4)$$

Equations (3.3.5) and (3.3.6) show the output of the attention block. The  $\alpha_{is}^t$  is multiplied with the input and added to the output.

$$g_i = g_i + \alpha_{ie}^t \cdot g_i \quad (3.3.5)$$

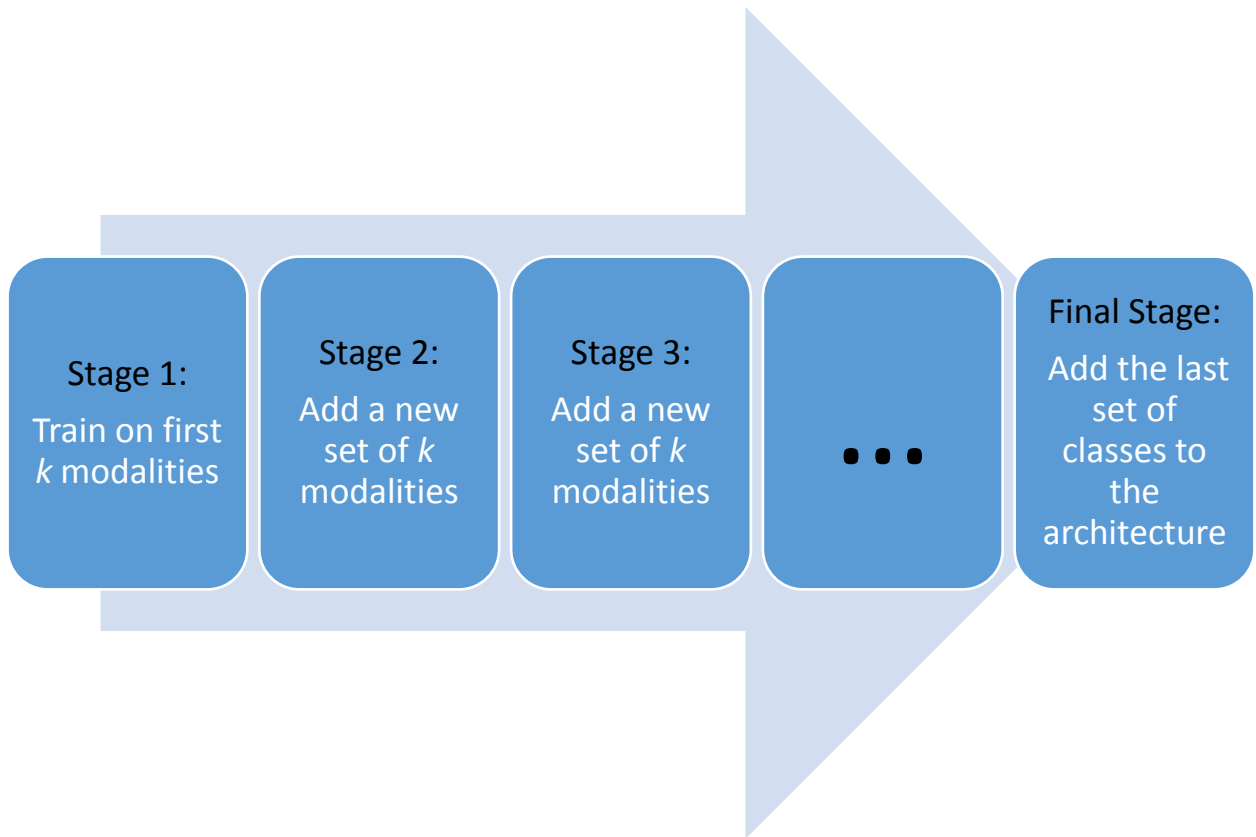
$$g_s = g_s + \alpha_{se}^t \cdot g_s \quad (3.3.6)$$

### 3.4 Stage Wise Learning for Multi Modal Embeddings

To overcome the limitations of the lesser number of multi-modal datasets, we develop a model that performs the stage-wise addition of information into the vector space. The subsequent addition of new information can be considered as the new modalities that are being added to the space. In the stage-wise learning method, we train our model on subsets of the dataset and then incrementally keep adding additional information into the model. The stage-wise learning technique can be thought of as a way of pre-training the network. For example, in the first stage, we add information about only 10 classes and train the model. In the subsequent stage, we use the model from the previous stage and add new class information to the network by training it for these additional new classes. We evaluate this model on two sets of data:

1. A subset of the test data which included only the information from the first stage of training.
2. The entire holdout test dataset.

In ‘1.’, we expect the recall scores to decrease across the stages whereas the model becomes more generalized. In the first stage, it would have overfitted on the initial classes it was trained on but as the stages progress, the overfitting problem will be resolved. In ‘2.’, the model will initially perform poorly on the entire test set because it has seen very few classes and the model is not robust. However, as the training stages progress, we can see it improving because as the model sees more classes, it starts performing better on the test dataset. We limit our work and experiments in this architecture to the image and text modalities only.



*Figure 17: Step by step implementation of stage-wise learning.*

### ***A. Training Strategy***

The training strategy is described in Figure 17. Consider a training set containing  $N$  categories. We divide these categories into groups of  $K$  categories each. In the first stage,  $K$  categories are trained. In the subsequent stages, further  $K$  categories are added to the network. At each stage, the loss is calculated with respect to the  $K$  categories that are being added.

### 4.1 Datasets

We use multiple multi-modal datasets to test our architecture. Commonly, most datasets consist of two modalities. We evaluate the CVS, M-CVS and RVS space on two multimodal datasets and the CVS on the two cross modal datasets.

#### *A. XMedia [11, 47] and XMediaNet [51,59]*

The XMedia dataset contains five different modalities of data. It contains images, text, audio, video and 3D point data. The number of samples in each of the modalities is stated in Table 1. XMedianet is an extension of the XMedia dataset which has the same modalities of data but with more data points. The numbers in Table 1 indicate the training and testing split for both the datasets. The XMedia and XMediaNet dataset come as pre-extracted features as shown in Table 2. The 200 categories in the XMediaNet dataset are classified into primarily two parts: animals and artifacts. While animals like elephant, owl, honey bees make up 48 out of the 200 categories, the artifacts make up the remaining 152 categories including things like violin, airplane, shotgun, camera, etc. Figure 18 and Figure 19 display samples from the XMedia and XMediaNet datasets respectively.








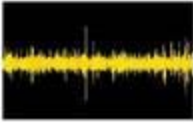















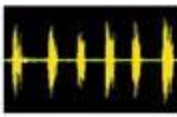

















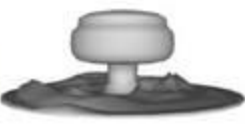
	Image	Text	Audio	Video	3D
<b>Laughter</b>		<p>Laughter is the natural and healthy response to the highest and noblest of the human emotions. It is a sign of the triumph of the good over the evil, of the light over the darkness, of the truth over the lies. It is a sign of the victory of the soul over the body, of the spirit over the flesh, of the mind over the senses. It is a sign of the triumph of the human over the animal, of the civilized over the savage, of the noble over the base. It is a sign of the triumph of the good over the evil, of the light over the darkness, of the truth over the lies. It is a sign of the victory of the soul over the body, of the spirit over the flesh, of the mind over the senses. It is a sign of the triumph of the human over the animal, of the civilized over the savage, of the noble over the base.</p>			
<b>Stream</b>		<p>The beautiful and peaceful stream flows through the lush green forest, its gentle murmur filling the air. The water is crystal clear, reflecting the sunlight and the surrounding trees. The sound of the water is a soothing melody, a reminder of nature's beauty and tranquility. The stream is a source of life and vitality, providing water for the plants and animals that call it home. It is a testament to the power of nature and the beauty of the natural world.</p>			
<b>Wolf</b>		<p>Wolf is a member of the Canidae family and is found in various parts of the world. It is a highly intelligent and social animal, known for its pack behavior. Wolves are skilled hunters and are often found in mountainous and forested areas. They are also known for their howling, which is a form of communication within the pack. The wolf is a symbol of strength, loyalty, and courage. It is a magnificent creature that has captured the imagination of humans for centuries.</p>			
<b>Airplane</b>		<p>A Boeing 737 is a narrow-body aircraft that is widely used for short-haul flights. It is a highly efficient and reliable aircraft, capable of carrying up to 189 passengers. The 737 is known for its maneuverability and fuel efficiency, making it a popular choice for airlines. It is a testament to the power of modern aviation and the ability of humans to travel long distances quickly and safely.</p>			
<b>Autobike</b>		<p>A Harley-Davidson is a brand of motorcycles that is known for its classic cruiser design. It is a highly reliable and durable motorcycle, capable of carrying up to two riders. The Harley-Davidson is a symbol of freedom and adventure, and is popular among riders of all ages. It is a testament to the power of the open road and the joy of riding.</p>			
<b>Bird</b>		<p>A small bird is perched on a branch, its feathers a mix of brown and white. It is looking towards the camera with a curious expression. The bird is a member of the Passeridae family and is found in many parts of the world. It is a highly intelligent and social animal, known for its ability to learn and mimic sounds. The bird is a symbol of freedom and flight, and is a beloved member of the avian world.</p>			
<b>Dog</b>		<p>A dog is a member of the Canidae family and is found in many parts of the world. It is a highly intelligent and social animal, known for its loyalty and companionship. Dogs are often used as working animals and are also popular as pets. They are a symbol of friendship and loyalty, and are a beloved member of the human family.</p>			
<b>Drum</b>		<p>A drum is a member of the membranophone family and is found in many parts of the world. It is a highly versatile and expressive instrument, capable of producing a wide range of sounds. Drums are often used in traditional music and are also popular in modern music. They are a symbol of rhythm and energy, and are a beloved member of the musical world.</p>			
<b>Elephant</b>		<p>An elephant is a member of the Proboscidea family and is found in many parts of the world. It is a highly intelligent and social animal, known for its strength and longevity. Elephants are often used as working animals and are also popular as pets. They are a symbol of wisdom and strength, and are a beloved member of the animal world.</p>			
<b>Explosion</b>		<p>An explosion is a sudden and violent release of energy, often resulting in the formation of a shockwave. It is a highly destructive and powerful event, capable of causing significant damage and loss of life. Explosions are often used in warfare and are also a common occurrence in nature. They are a symbol of power and destruction, and are a feared member of the natural world.</p>			

Figure 18: Example from the XMedia [11, 47] dataset.

Table 1: XMedia and XMediaNet dataset statistics. Figures indicate the number of training and testing samples.

	XMedia	XMediaNet
<b>Image</b>	4000, 1000	32000, 8000
<b>Text</b>	4000, 1000	32000, 8000
<b>Video</b>	400, 100	8000, 2000
<b>Audio</b>	800, 200	8000, 2000
<b>3D</b>	400, 100	1600, 400
<b>Categories</b>	20	200






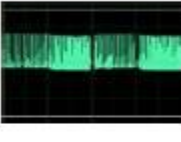














	Image	Text	Audio	Video	3D
<b>violin</b>		<p>the most commonly available stringed instrument, a standard violin is manufactured by hand under the supervision of a master craftsman. The instrument is commonly used in orchestras and chamber music ensembles. It is a transverse instrument, meaning that the player holds it horizontally and uses the bow to vibrate the strings. The body is made of wood and is relatively small.</p>			
<b>flute</b>		<p>The alto flute is a type of Western concert flute, a musical instrument in the woodwind family. It is the next lowest pitched of the C flute family. The alto flute is characterized by its distinct, mellow tone in the lower portion of its range. It is a transverse instrument in its design, like the piccolo and bass flute, and uses the same fingering as the C flute.</p>			
<b>airplane</b>		<p>Airbus began as a consortium of aerospace manufacturers. Airbus Industrie, a consortium of European defence and aerospace companies, was established in 1980 and was the first of a number of companies to be established in 2000, owned by the European Aerospace Defence and Space Company (EADS), and BAE systems (BAE).</p>			
<b>helicopter</b>		<p>The Apache originally came in the Model 77 developed by Hughes (acquired by the United States Army's Advanced Aircraft Development program to replace the AH-1 Cobra). The prototype, Model 77, was built by Hughes in 1975. The Army selected the Model 77 for development in 1976, and later approved full production in 1982.</p>			
<b>camera</b>		<p>A camera is an optical instrument for recording or capturing images, which may be stored locally, transmitted to another location, or both. The images may be individual or sequential, or sequential images may be recorded on a single strip of film. The camera is a device that captures images of scenes or objects without physical contact.</p>			

Figure 19 Example from the XMediaNet [51, 59] dataset.

Table 2: XMedia and XMediaNet Features and their dimensions.

	XMedia	XMediaNet	Dimensions
<b>Image</b>	CNN	CNN	4096
<b>Text</b>	Bag of Words	Bag of Words	2048
<b>Video</b>	CNN	CNN	4096
<b>Audio</b>	MFCC	MFCC	29
<b>3D</b>	LightField	LightField	4700

### B. Nuswide [49] and Pascal [50]

The Nuswide and Pascal datasets contain images and their corresponding text data. Each of the images and text samples belongs to one category and we perform retrieval for an input query image or text. Figure 20 displays samples from the Nuswide dataset and Table 3 describes the dataset statistics for the Nuswide and Pascal datasets.



Figure 20: Sample images from the Nuswide dataset. [49]

### C. Birds [58]

We show zero-shot retrieval on the Birds dataset to evaluate the performance of the stage-wise learning model. The Birds dataset contains 150 categories for training and 50 unseen categories for testing. The goal of the zero-shot retrieval is to understand if the model is able to perform well on unseen categories and if the clusters that are being formed are robust or not. For the stage-wise learning methods, we split the Pascal and Nuswide datasets into groups of five classes and implement stage-wise learning on these groups. Additionally, we also report

stagewise learning scores for the zero-shot learning on the Birds dataset. Table 4 describes the dataset statistics for the Birds.

Table 3: Pascal and Nuswide dataset statistics. Figures indicate the train and test splits.

	<b>Pascal</b>	<b>Nuswide</b>
<b>Image</b>	800, 200	8000, 2000
<b>Text</b>	800, 200	8000, 2000
<b>Categories</b>	10	20

D.

Table 4: Train and test split for the Birds dataset used to perform zero-shot retrieval.

	<b>Birds</b>
<b>Train</b>	8855, 2933
<b>Categories</b>	150, 50

## 4.2 Implementation

We implement cross-modal retrieval on these datasets using gradient descent based techniques where the goal is to minimize the loss functions described in Section 3. We pre-extract the vector representations of all our data. We use ResNet architecture to extract the image features and SkipThoughts to extract the text features for the cross modal retrieval task. For the multimodal retrieval task, we use the features enlisted in Table 2. We use Tensorflow on Nvidia GPUs to perform all our experiments and use a 512 dimension common embedding space. Our batch size is fixed at 128 with a learning rate of 0.001. For the triplet loss, we set all the margins ( $\alpha_1, \alpha_2, \alpha_3$ ) to 1.0. We set  $\gamma_1, \gamma_2, \gamma_3$  to 0.25 and 0.25, 0.5 respectively.

### A. Evaluation Metric

Precision, recall, F1 score, accuracy are some of the most popular model evaluation metrics. In a task like retrieval, we want to measure the precision of our model for every query. In case of multiple queries, we want to define a metric that that represents the performance of the model for all the queries. That is why, mean average precision is the chosen metric in the area of information retrieval.

We calculate the average precision value for the top  $K$  queries.

$$AP = \frac{1}{N} \sum_{n=1}^K (p(r) \cdot rel(r)) \quad (4.2.1)$$

Where:

- $N$  is the number of relevant data samples in the retrieved results.
- $p(r)$  is the precision at  $r$  and  $rel(r)$  is a flag that indicates if the retrieved result is a match or not.

The mAP is obtained by averaging the AP of all the queries. We report the *mAP at 50* ( $K = 50$ ) on all our experiments.

### B. Intuitive Understanding of mAP

Retrieval is primarily the task of searching for information that is very similar to the query. Figure 21 depicts an intuitive way of understanding how this metric is calculated. Mean average precision is a standard metric used to measure the performance of a retrieval system. Let  $Q$  be a user query,  $G$  be a set of labeled data in our common embedding space and  $d(i,j)$  be a measure of the similarity between two objects  $i,j$ . Let  $G'$  be the ordered set of  $G$  according to the function  $d(i,j)$  and  $k$  be an index of  $G'$ .

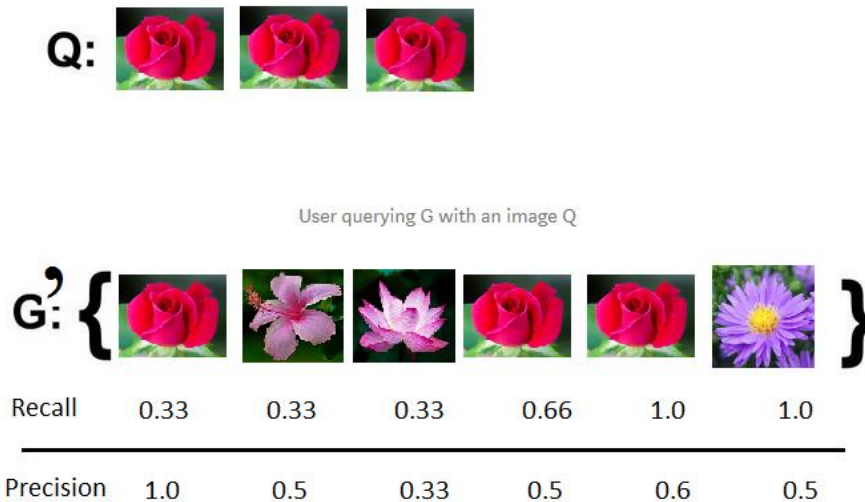


Figure 21: A visualization of mean average precision.

In the above query, the average precision for the  $Q$  is given by:

$$AP = \frac{(1.0 + 0.5 + 0.6)}{3}$$

Similarly, the AP is calculated for the queries in the test set and mAP is the average of all the APs.

## 5.1 Results

In this section, we compare all the implementations discussed in Chapter 3 on the datasets discussed in Chapter 4. We report mAP@50 scores and different visualizations to analyze our results.

### A. CVS Model: Pascal and Nuswide

We evaluate our CVS model on the Pascal and the Nuswide datasets. We observe that the addition of the aligned attention mechanism improves retrieval as is evident from the scores. The aligned attention gives an improvement of 15% over the baseline CVS model.

Table 5: mAP scores for the Pascal and Nuswide dataset.

Dataset	Method	img2txt	txt2img
Pascal	UNSCM	0.304	0.282
	Deep-SM	0.446	0.478
	ACMR	0.535	0.543
	DCKT	0.582	0.587
	MCSM	0.598	0.597
	CBT	0.602	0.583
	DSCMR	0.710	0.722
	<b>Baseline CVS</b>	<b>0.591</b>	<b>0.567</b>
	<b>CVS + Aligned Attention</b>	<b>0.660</b>	<b>0.671</b>
Nuswide	UNSCM	0.312	0.354
	CCL	0.506	0.535
	CSGH	0.542	0.569
	ACMR	0.544	0.538
	SC ACMR	0.545	0.448
	DCKT	0.556	0.584
	UGACH	0.631	0.641
	DSCMR	0.611	0.615
	<b>Baseline CVS</b>	<b>0.450</b>	<b>0.485</b>
	<b>CVS + Aligned Attention</b>	<b>0.574</b>	<b>0.676</b>

### B. CVS Model: XMedia and XMediaNet (2 modalities)

Table 6 displays the scores on the XMedia and XMediaNet when it is trained only on image and text modalities.

Table 6: mAP scores for the Xmedia and XMediaNet dataset.

Dataset	Method	img2txt	txt2img
Xmedia	CBT	0.516	0.464
	CM-GAN	0.567	0.551
	Baseline CVS	0.895	0.902
	CVS + Aligned Attention	0.908	0.950
XmediaNet	CBT	0.516	0.464
	CM-GAN	0.567	0.551
	Baseline CVS	0.536	0.495
	CVS + Aligned Attention	0.598	0.583

### C. CVS Model: Xmedia and XMediaNet (5 modalities)

Table 7 represents the scores on the XMediaNet when it is trained on all the modalities. The scores for images and sentences are much higher because of the higher number of training samples in each of these modalities. The larger corpus of training data for these two modalities means that the model is able to generalize well across a large spectrum of data. The same cannot be said about videos as it has fewer samples to train on (see Table 2).

Table 7: mAP scores for the images, text, and video for the XMediaNet dataset.

Attention		I	S	V
No	I		0.688	0.417
	S	0.616		0.347
	V	0.306	0.281	
	Average	<b>0.442</b>		
Yes	I		0.775	0.605
	S	0.685		0.504
	V	0.395	0.383	
	Average	<b>0.558</b>		

Table 8 shows the scores on the XMedia when it is trained on all the modalities. Similar to the XMediaNet dataset, the scores of images to sentence and sentence to image retrieval are high owing to the vast amount of training data that is present in these two modalities.



Table 8: Cross modal retrieval scores (mAP) for the five modalities in the XMedia dataset.

Method	Q → R	I	T	A	V	3D
CVS	I	-	0.895	0.759	0.606	0.624
	T	0.902	-	0.763	0.550	0.637
	A	0.504	0.480	-	0.293	0.446
	V	0.138	0.317	0.253	-	0.153
	3D	0.436	0.580	0.412	0.129	-
	Avg	<b>0.494</b>				
CVS + Attention	I	-	0.908	0.708	0.801	0.731
	T	0.95	-	0.743	0.828	0.769
	A	0.416	0.477	-	0.341	0.42
	V	0.49	0.481	0.366	-	0.434
	3D	0.58	0.545	0.457	0.558	-
	Avg	<b>0.600</b>				

To coalesce the twenty different retrieval scores, we evaluate a model based on the average performance across all these modalities. We observe that the CVS model with the aligned attention works best and gives an average retrieval score of 0.6. The 3D and audio modalities have fewer samples and low dimensionality respectively and hence the scores for these two modalities are comparatively lower. Figure 22 describes the decrease in the total loss as training progresses.

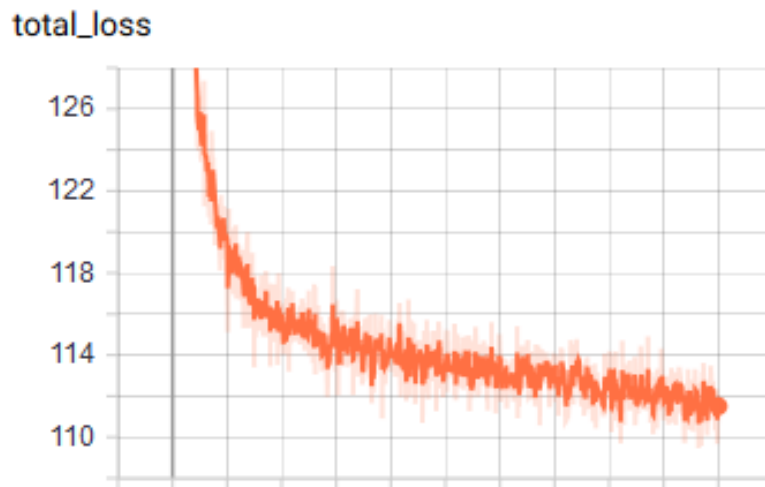


Figure 22 The total loss for the CVS model as a function of the epochs visualized on Tensorboard.

#### D. M-CVS Model: Xmedia and XMediaNet (5 modalities)

In the M-CVS model, we implement a four-stage learning approach for the XMedia dataset. The four stages help build the model such that each new stage does not affect previous stages of training. The four stages of training are described below.

- Stage 1: Train images and text modalities.
- Stage 2: Perform incremental addition of the video modality.
- Stage 3: Add audio modality to the architecture.
- Stage 4: Add 3D modality to the stage-wise learning architecture.

The scores reported in Table 9 are the scores after the fourth stage of training. These scores are comparable with the CVS model.

Table 9: Cross modal retrieval scores(mAP) for the M-CVS model on the XMedia dataset.

Method	Q -> R	I	T	A	V	3D
S2UPG	I	-	0.27	0.265	0.264	0.394
	T	0.275	-	0.242	0.242	0.338
	A	0.274	0.244	-	0.207	0.363
	V	0.225	0.193	0.168	-	0.267
	3D	0.345	0.275	0.329	0.276	-
	Avg	0.273				
SCVM	I	-	0.903	0.406	0.532	0.655
	T	0.889	-	0.507	0.549	0.722
	A	0.438	0.527	-	0.313	0.37
	V	0.553	0.58	0.302	-	0.45
	3D	0.603	0.679	0.37	0.426	-
	Avg	0.539				
M-CVS	I	-	0.897	0.531	0.809	0.782
	T	0.943	-	0.538	0.840	0.822
	A	0.508	0.513	-	0.202	0.505
	V	0.521	0.211	0.318	-	0.282
	3D	0.616	0.627	0.338	0.537	-
	Avg	0.567				

In the XMediaNet dataset, we perform just a 2 stage of training for the training. In stage 1, we train image and text modalities and then add the video modality. While this model doesn't perform as well as the CVS model, it still performs better than the CVS model without the aligned attention

mechanism. Figure 23 shows the decrease in the metric loss across subsequent epochs for the M-CVS model.

Table 10: mAP scores for images, text, and video for XMediaNet dataset.

	<b>I</b>	<b>S</b>	<b>V</b>
<b>I</b>	-	0.777	0.457
<b>S</b>	0.673	-	0.349
<b>V</b>	0.227	0.260	-

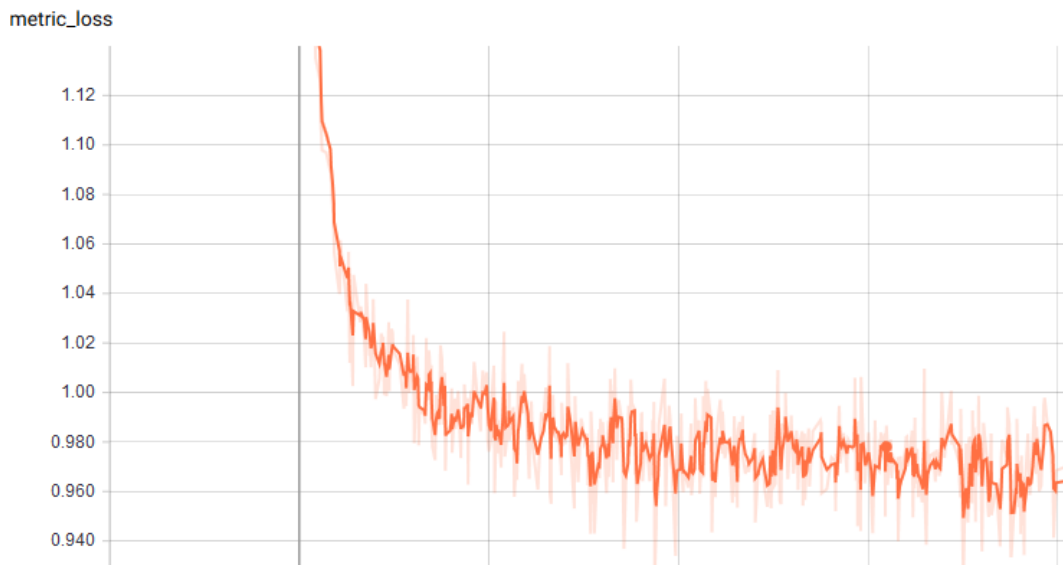


Figure 23: The metric loss for the M-CVS model with respect to the total epochs.

### ***E. RVS Model: Xmedia and XMediaNet (5 modalities)***

Similar to the CVS and the M-CVS, we perform stage-wise training, but now introduce a reference modality and train all other modalities with respect to that modality. For example, we assign image as a reference modality and in stages 1, 2, 3 and 4, we train this reference with text, audio, video and 3D point clouds respectively.

We evaluate the RVS architecture on the XMedia and XMediaNet dataset and observe that the architecture remains highly constrained because of the lack of the update of the parameters. Since the RVS contains only one attention block in each subsequent stage of training, our model doesn't generalize to all modalities across all the classes.

In our setting, we set the image branch to be the reference modality and the scores are shown in Table 11. If we change the reference modality to the text modality, we observe a slight increase in

the average score as shown in Table 11. However, in comparison with Tables 8 and 9, we can observe that the RVS doesn't perform as well as the CVS or the M-CVS model. This can be attributed to the fact that the CVS and the M-CVS models are more flexible and have more modality-specific parameters than the RVS model. Figure 24 shows the decrease in category loss across epochs for the RVS architecture.

We experiment with changing the reference modality to text and as shown in Table 11.

Table 11: Cross modal retrieval mAP scores for the XMedia dataset.

Method	Q → R	I	T	A	V	3D
RVS (image is reference)	I	-	0.909	0.439	0.806	0.208
	T	0.945	-	0.148	0.090	0.095
	A	0.285	0.071	-	0.147	0.155
	V	0.539	0.078	0.185	-	0.143
	3D	0.116	0.045	0.134	0.147	-
	Avg	<b>0.284</b>				
RVS (text is reference)	I	-	0.942	0.610	0.844	0.265
	T	0.902	-	0.136	0.074	0.185
	A	0.047	0.051	-	0.144	0.155
	V	0.535	0.113	0.177	-	0.145
	3D	0.085	0.118	0.134	0.174	-
	Avg	<b>0.291</b>				

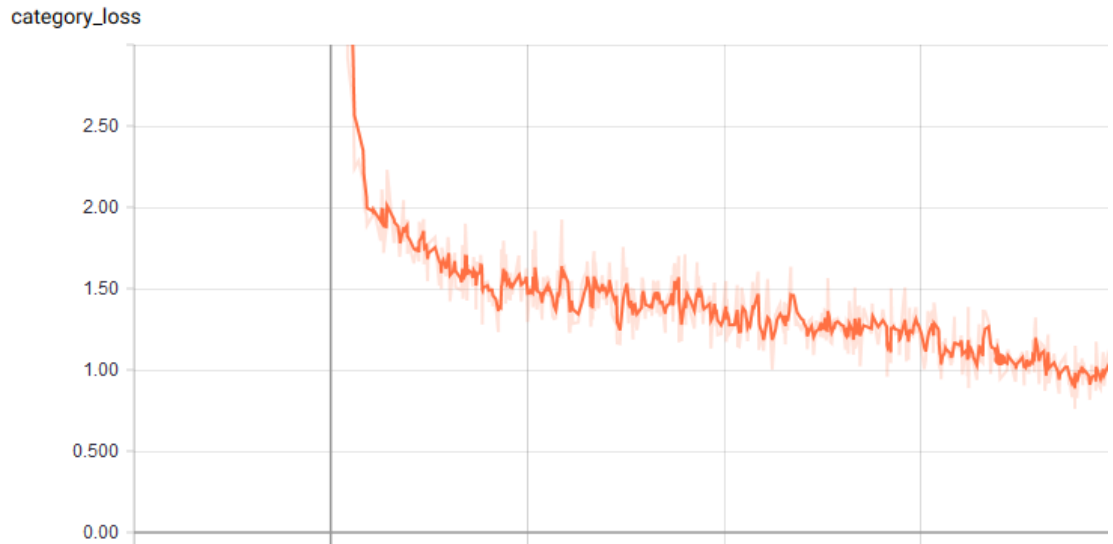


Figure 24: The decrease in the cross entropy loss with respect to the total number of epochs in the RVS model.

## F. Stagewise Learning

The stage-wise learning model is evaluated on the Pascal, Nuswide and Birds dataset. In the stage-wise learning for these datasets, we consider a setting where different classes are introduced into the model at different stages. For comparing the stage-wise learning technique, we evaluate our model on two test sets. The first is the holdout test set that belongs to the classes that were introduced in the first stage and the second is to test our model on the entire held out test set. This allows comparison of two things: 1) sensitivity to addition of new data to the model, and 2) the robustness of our model to be able to retain what is learned in a previous stage. In this experiment, we incrementally add new classes to the model at each stage but do not fine-tune the model for the previously existing stages of training. For instance, in the Pascal dataset, we try the following stage-wise training technique. Table 12 describes the method of splitting the datasets for training across multiple stages.

Table 12: Stage wise learning methodology for the Pascal and Birds dataset.

Stage	Train on classes	Eval on (Part 1) classes	Eval on (Part 2)
1	1-5	Test samples from 1-5	Entire test set
2	6-10	Test samples from 1-5	Entire test set
3	11-15	Test samples from 1-5	Entire test set
4	15-20	Test samples from 1-5	Entire test set

Using the above technique, we observe that the retrieval for the classes that were trained in the first stage decrease over time as shown in Table 13. This is expected because as we introduce more classes into the space, we expect the model to perform a little worse on the data that was present in the first stage of training in exchange for better generalization. Similarly, we observe that the retrieval scores on the entire holdout test set start increasing as we increase more class information into the space across the four stages. However, this sort of training technique is still not efficient as compared to the CVS model where we directly train all the data points together.

On the zero shot retrieval on the Birds dataset, we observe that the model gets stuck in a local minima and fails to learn across stages as can be seen in the decrease in scores on the holdout test set across the four stages.

Table 13: Stage-wise learning retrieval scores on Pascal and Birds dataset.

Dataset			5 Categories	10 Categories	15 Categories	20 Categories	CVS
Pascal	Eval On		Stage1	Stage 2	Stage 3	Stage 4	-
	5 categories from Stage 1	i2t	0.857	0.836	0.844	0.810	-
	5 categories from Stage 1	t2i	0.903	0.889	0.895	0.845	-
	All 20 categories	i2t	0.385	0.399	0.401	0.446	0.639
	All 20 categories	t2i	0.420	0.470	0.490	0.500	0.650
Dataset			40 Categories	40 Categories	40 Categories	30 Categories	CVS
Birds	Eval On		Stage1	Stage 2	Stage 3	Stage 4	-
	5 categories from Stage 1	i2t	0.423	0.419	0.378	0.376	-
	5 categories from Stage 1	t2i	0.287	0.282	0.273	0.277	-
	All 50 categories	i2t	0.349	0.365	0.341	0.361	0.538
	All 50 categories	t2i	0.248	0.247	0.238	0.237	0.589

## 5.2 Ablation Analysis

### A. Attention Ablation Analysis

The aligned attention consists primarily of three weight matrices:  $W_i$ ,  $W_s$  and  $W_{is}$ . The  $W_i$  and  $W_s$  cater to the intramodality attention alignment and  $W_{is}$  caters to the intermodality alignment. Our experiments show that both the intermodality and intramodality attention mechanisms are important and not having all three weight matrices in the attention mechanism degrades the scores significantly. Figure 25 shows the effect of eliminating these weight matrices on the results. The scores for the retrieval significantly reduce if we omit either of the weight matrices. This shows that our attention mechanism does reasonably well in minimizing the loss within the same modality and also across two modalities.

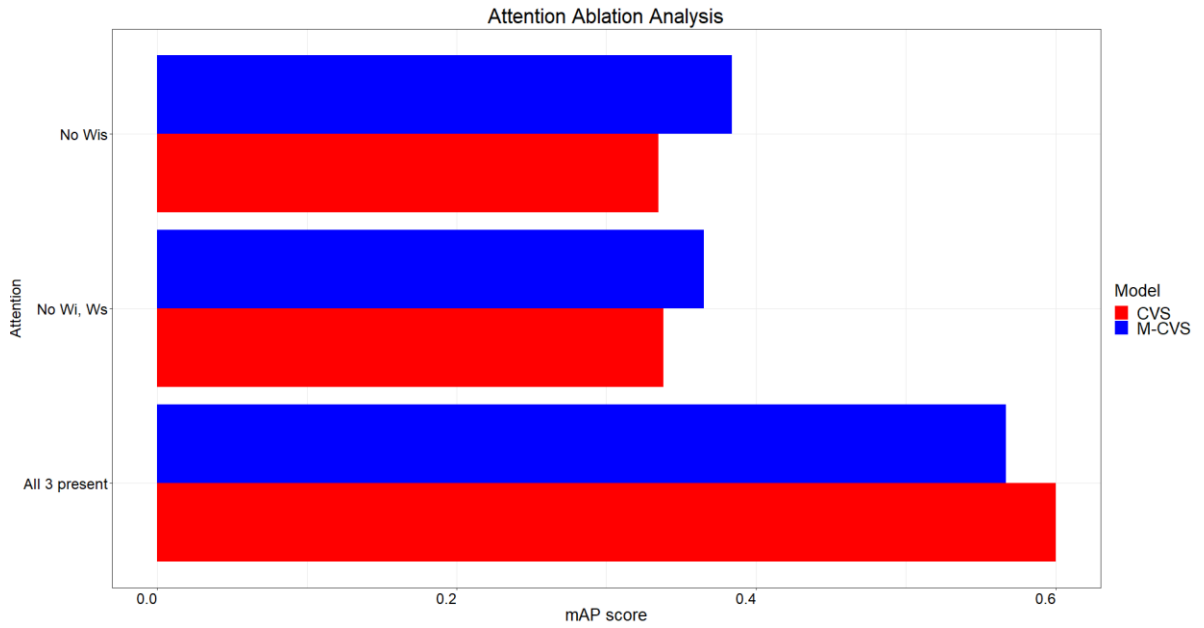


Figure 25 Effect of all the 3 components of attention.

Figure 26 depicts the increase in total attention blocks as we keep adding more modalities to our model. The CVS model will need to train a lot more attention modules when compared to the M-CVS and the RVS model.

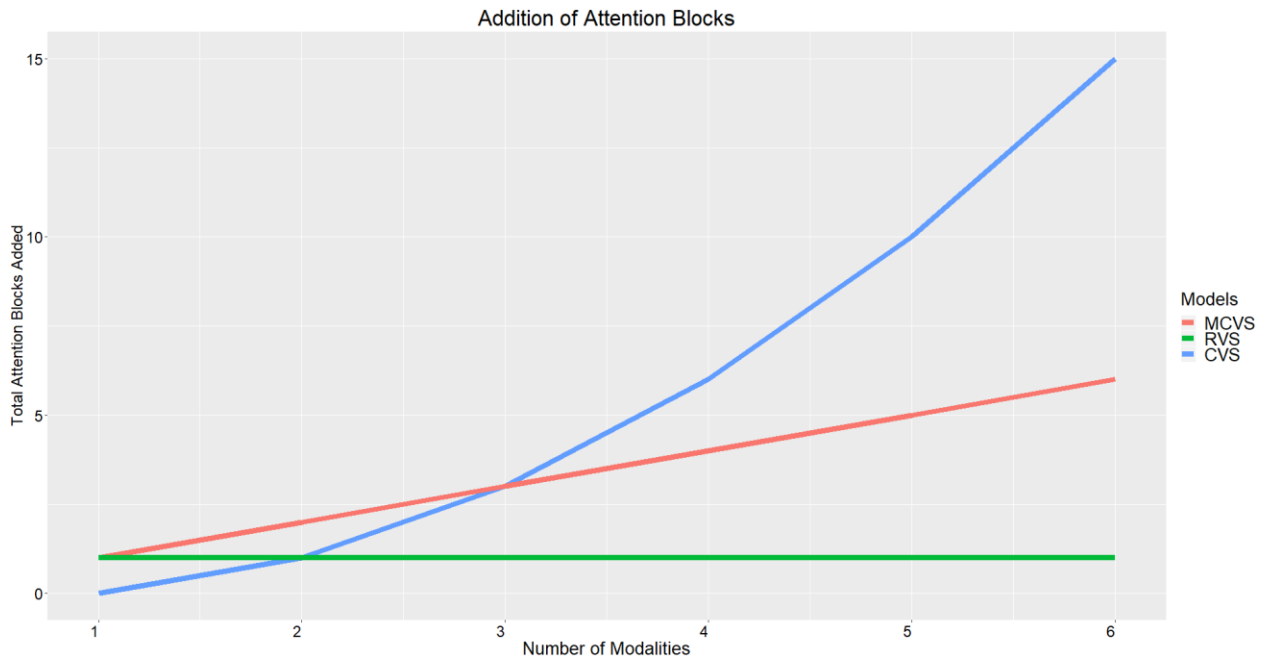


Figure 26: The total attention blocks that need to be trained when we add a new modality.

## B. Timing Analysis

Figure 27 indicates the time taken per iteration for the CVS and the M-CVS model to train. The CVS model restarts training from scratch every time a new modality is added and it is shown in the extra time that the model takes per iteration as it has to update more parameters. On the other hand, in the M-CVS model, we can observe that the time at each subsequent stage is lesser because only the new parameters of the new modality network branch are updated and the existing networks remain unchanged.

The additional time taken to train a new modality is a function of the size of the training data and the size of the feature vector. The time taken by the M-CVS model to go from three to four modalities is less in terms of the milliseconds per iteration. This is because the fourth modality that is added to the model is the audio branch and this branch has only 29 dimensions in the XMedia dataset. Owing to such low dimensionality, the total number of parameters in the fully connected layers is much lesser when compared to the other modalities and hence we can't see such a small increment in the time taken during training.

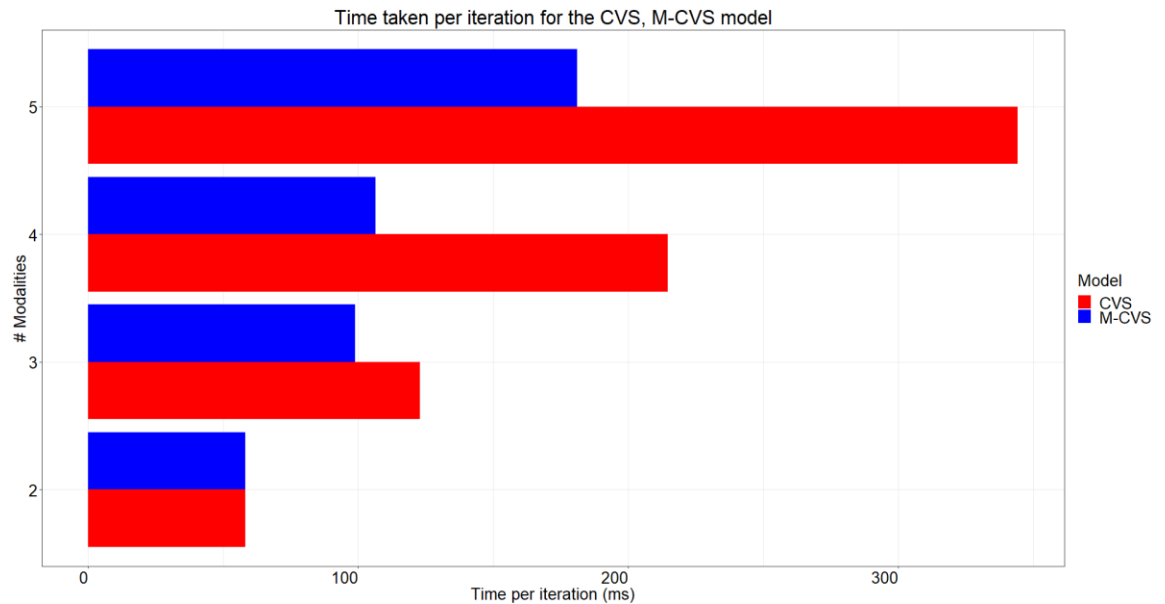


Figure 27: The time per iteration for the CVS and the M-CVS model as we increase the number of modalities added to the model.

## C. Adversarial Loss Functions

We attempted to include image reconstruction loss and sentence reconstruction loss into our model. The objective of this loss function was to reconstruct the 512-dimensional input vector



representation. It was observed that the addition of such a loss function was not very helpful to the model as indicated in Figure 28. The loss value explodes and our model does not converge. However, if we include the metric losses and the classifier in our model, we are able to still perform retrieval.



*Figure 28: Non converging of the reconstruction loss for the image and text modality.*

### 6.1 Conclusions

We present a technique for the transfer of information from one modality of data to another by using various common embedding spaces. This work provides an analysis of the different ways in which one can train a multi-modal neural network architecture and measure its performance. The methods discussed in this work can be used at different stages of learning or can also be used as an initializing for a multi-modal network that can be fine-tuned later.

### 6.2 Discussions

To summarize our findings,

- The CVS model can be used to train architectures where the entire data that needs to be projected is available beforehand.
- Adding any new modality into the CVS model at a later stage would require re-training the entire model from scratch.
- To avoid this problem, we use M-CVS model that performs stage-wise addition of modalities into the embedding space.
- This helps preserve existing transformations while being able to robustly add new modalities.
- Stage wise learning technique is promising but fails to generalize as well as the CVS model.
- We introduce a RVS model that uses one of the modalities (image and text were both explored) as the common embedding from which all other modalities must map to.

### 6.3 Future Work

The primary research of this thesis work was based on search and retrieval applications. The datasets available for the multi-modal networks are limited in number and have a lot of discrepancies. The model we build is not trained end-to-end. Some possible directions for extending this work are:

- Expand the scope of the network to generate samples of one modality from another using generative adversarial networks. For example: given an input image, we can make our model produce audio, video, 3D point cloud or a text using the common embedding space. This can be done using some sort of adversarial training.
- Implement end-to-end training of the model where the encoder layers are not frozen and we can update the weights in those layers as we learn the embedding weights. This can help improve our results but will come at a computational cost.
- Test our model on more multi-modal datasets with access to the raw data to perform end to end training including using multiple layers from the CNN, RNN, etc. to extract features instead of simply relying on the final fully connected layer of the neural network.

# Bibliography

---

- [1] L. Wang, Y. Li, J. Huang, and S. Lazebnik, “Learning Two-Branch Neural Networks for Image-Text Matching Tasks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 394–407, Feb. 2019.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
- [4] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015
- [5] C. Szegedy *et al.*, “Going Deeper With Convolutions,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
- [7] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv:1207.0580 [cs], Jul. 2012.
- [8] Jinwei Qi, Xin Huang, and Yuxin Peng, “Cross-media similarity metric learning with unified deep networks,” *Multimedia Tools and Applications*, vol. 76, no. 23, pp. 25109–25127, Dec. 2017.
- [9] Jinwei Qi and Yuxin Peng, “Cross-modal bidirectional translation via reinforcement learning.” in *IJCAI*, 2018, pp. 2630–2636.
- [10] Aviv Eisenschat and Lior Wolf, “Linking image and text with 2-way nets,” in arXiv preprint arXiv: 1608.07973, 2016.

- [11] Xiaohua Zhai, Yuxin Peng, and Jianguo Xiao, "Learning cross-media joint representation with sparse and semi supervised regularization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 965–978, 2014
- [12] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823
- [13] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015, pp. 3294–3302
- [14] Jeffrey Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015, pp. 2625–2634
- [15] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Rich Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057
- [16] S. Sah, S. Kulhare, A. Gray, S. Venugopalan, E. Prud'Hommeaux, and R. Ptucha, "Semantic Text Summarization of Long Videos," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 989–997.
- [17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [18] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [19] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten, "Densely connected convolutional networks," *arXiv preprint arXiv:1608.06993*, 2016.
- [20] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan. Cross-modal retrieval with cnn visual features: A new base-line. *IEEE transactions on cybernetics*, 47(2):449–460, 2017.
- [21] J. Qi, X. Huang, and Y. Peng. Cross-media similarity metric learning with unified deep networks. *Multimedia Tools and Applications*, 76(23):25109–25127, Dec. 2017.
- [22] F. Feng, X. Wang, and R. Li. Cross-modal retrieval with correspondence autoencoder. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 7–16. ACM, 2014.
- [23] X. Huang and Y. Peng. Deep cross-media knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8837–8846, 2018.

- [24] Vendrov, Ivan, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. "Order-embeddings of images and language." arXiv preprint arXiv:1511.06361 (2015)
- [25] You, Quanzeng, Zhengyou Zhang, and Jiebo Luo. "End-to-End Convolutional Semantic Embeddings." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5735-5744. 2018.
- [26] Xin Huang and Yuxin Peng, "Deep cross-media knowledge transfer," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp.8837–8846.
- [27] Lin Wu, Yang Wang, and Ling Shao, "Cycle-consistent deep generative hashing for cross-modal retrieval," IEEE Transactions on Image Processing, vol. 28, no. 4, pp. 1602–1612, 2019.
- [28] Suris, D., Duarte, A., Salvador, A., Torres, J., Giró-i Nieto, X.: Cross-modal embeddings for video and audio retrieval. arXiv preprint arXiv:1801.02200 (2018)
- [29] W. Havard, L. Besacier, and O. Rosec, "SPEECH-COCO: 600k Visually Grounded Spoken Captions Aligned to MSCOCO Data Set," arXiv:1707.08435 [cs], Jul. 2017.
- [30] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," arXiv:1405.0312 [cs], May 2014.
- [31] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
- [32] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 568–576.
- [33] W. Shi *et al.*, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 1874–1883.
- [34] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," pp. 850–855, Jan. 1999.
- [35] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.
- [36] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu. Deepvisual-semantic hashing for cross-modal retrieval. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1445–1454, New York, NY, USA, 2016. ACM.

- [37] F. Zheng, Y. Tang, and L. Shao. Hetero-manifold regularization for cross-modal hashing. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1059–1071, 2018
- [38] Hadsell, Raia, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping." *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. Vol. 2. IEEE, 2006.
- [39] P. Anderson *et al.*, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6077–6086.
- [40] P. Anderson *et al.*, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6077–6086.
- [41] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked Cross Attention for Image-Text Matching," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 201–216.
- [42] X. Xu, L. He, H. Lu, L. Gao, and Y. Ji, "Deep adversarial metric learning for cross-modal retrieval," *World Wide Web*, vol. 22, no. 2, pp. 657–672, Mar. 2019.
- [43] L. Zhen, P. Hu, X. Wang, and D. Peng, "Deep Supervised Cross-Modal Retrieval," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10394–10403.
- [44] Y. Fang, H. Zhang, and Y. Ren, "Unsupervised cross-modal retrieval via Multi-modal graph regularized Smooth Matrix Factorization Hashing," *Knowledge-Based Systems*, vol. 171, pp. 69–80, May 2019.
- [45] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-Based Deep Hashing Network for Cross-Modal Retrieval," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3893–3903, Aug. 2018.
- [46] L. Wu, Y. Wang, and L. Shao, "Cycle-Consistent Deep Generative Hashing for Cross-Modal Retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1602–1612, Apr. 2019.
- [47] Y. Peng, X. Zhai, Y. Zhao, and X. Huang. Semi-supervised cross-media feature learning with unified patch graph regularization. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(3):583–596, 2016.

- [48] X. Zhai, Y. Peng, and J. Xiao. Learning cross-media joint representation with sparse and semi supervised regularization. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6):965–978, 2014.
- [49] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of Singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., July 8-10, 2009.
- [50] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon’s mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 139–147. Association for Computational Linguistics, 2010.
- [51] Y. Peng, X. Huang, and Y. Zhao. An overview of cross-media retrieval: Concepts, methodologies, benchmarks and challenges. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017
- [52] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.
- [53] Hardoon, David R., Szedmak, Sandor R., and Shawe-taylor, John R. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16:2639–2664, 2004.
- [54] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep Boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.
- [55] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, “Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models,” p. 9.
- [56] Y. He, S. Xiang, C. Kang, J. Wang, and C. Pan, “Cross-Modal Retrieval via Deep and Bidirectional Representation Learning,” *IEEE Transactions on Multimedia*, vol. 18, no. 7, pp. 1363–1377, Jul. 2016.
- [57] I. Goodfellow *et al.*, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.
- [58] Welinder P., Branson S., Mita T., Wah C., Schroff F., Belongie S., Perona, P. “Caltech-UCSD Birds 200”. California Institute of Technology. CNS-TR-2010-001. 2010.



- [59] Li, Cheng & Rana, Santu & Phung, Dinh & Venkatesh, Svetha. (2015). Data Clustering Using Side Information Dependent Chinese Restaurant Processes. Knowledge and Information Systems. 10.1007/s10115-015-0834-7.
- [59] Y. Peng, J. Qi and Y. Yuan, "Modality-specific Cross-modal Similarity Measurement with Recurrent Attention Network", IEEE Transactions on Image Processing (TIP), Vol. 27, No. 11, pp. 5585-5599, Nov. 2018.
- [60] C. Zhang, S. Sah, T. Nguyen, D. Peri, A. Loui, C. Salvaggio, and R. Ptucha. Semantic sentence embeddings for paraphrasing and text summarization. In 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 705–709. IEEE, 2017.
- [61] S. Sah, C. Zhang, T. Nguyen, D. K. Peri, A. Shringi, and R. Ptucha. Vector learning for cross domain representations. In 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pages 1–5. IEEE, 2017.
- [62] Understanding LSTMs <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>