Image-Text Combined Embedding Using Variable Cross-View and Within-View Constraints

Ridham Dave
Electrical and Computer Engineering
University of Waterloo
Canada

Vidya shree S
Electrical and Computer Engineering
University of Waterloo
Canada

Rithika Shivakumar
Computer Science
University of Waterloo
Canada

***Abstract*—Combining Visual and Textual domains into a common space has been an exciting research domain. In this work, we implement a method to learn image-text joint embeddings, where two classification models, separate for each domain, are combined to form a common representation. These classification models are further combined and trained using multi-component loss functions, where cross-view and within-view constraints are added to optimize Bi-directional ranking and to preserve structure between matching pairs. We experiment with margin values for such constraints and weight multipliers of individual loss components. Our results reveal that faster training times can be achieved for such algorithms by using variable margins and weight multipliers.**

***Index Terms*—Image-Text Embedding, Image-Text Search, Cross-View Ranking Constraints, Within-View Neighborhood Structure Preservation Constraints**

I. Introduction

The growth of digital content on the web has led to the exponential production of data in images, text, audio, and video content. With the expansion of multi-modal data, it is more difficult for users to retrieve the information they are interested in efficiently and accurately [1]. Most retrieval methods so far are based on a single modality, such as searching for articles by text, searching for images by images, or multi-modal search on the surface [2].

Cross-modal retrieval enables flexible retrieval across different modalities (e.g., texts vs. images and vice versa). It takes one type of data as the query to retrieve relevant data of another type. For example, a text query is given to a model which provides several similar images relevant to the query and also similar text related to itself [3]. The challenge of cross-modal retrieval is threefold. It includes effective integration of multi-modal information, the identification and extraction of complementary and discriminatory features, and the measurement of content similarity between heterogeneous data [4].

In the visual domain, the technique of comparing the raw pixels of a high-resolution input picture does not prove to be effective or efficient [5]. However, extracting lower-dimensional feature vectors for the image provides an indication of what the image represents, where such representation can be used for such intelligent tasks. Deep transfer learning techniques have gained substantial momentum in the computer vision community [6], [7]. First, a deep convolutional neural network (CNN) is trained on a large labelled dataset [8]. Then the convolutional layers are used as mid-level feature extractors on a variety of computer vision tasks [9]. This type of feature vector can effectively represent the image in search or similarity-matching tasks.

Similarly, text data in raw format is difficult for computers to process due to its semantic meaning and complex relationships. Currently, there are several methods available to help identify semantic similarities between two words. GloVe stands for Global vectors and is a word embedding method based on the co-occurrence statistics of a corpus [10]. Its popularity is attributed to its high performance in capturing the semantic similarity between two words and relatively low computational cost. Such representation works on the principle of creating a co-occurrence matrix using the probability of a recurring word in the corpus [11].

Image and text embeddings are numerical representations of real-world objects that translate high dimensional space to a lower one. The vector space quantifies the semantic similarity between categories where vectors close to each other exhibit similar characteristics compared to the ones afar.

Canonical Correlation Analysis is a popular approach for obtaining image-text joint embeddings [12]. This approach tries to find the linear combinations of two different datasets such that the correlation is maximum between them. It is efficient with the image and text features but comes with a high memory cost, as the computations involve loading all data into memory and computing co-variance between image and text data [13].

In the original work, authors [13] designed an objective function with two aspects, bi-directional ranking and structure-preserving constraints [13] to establish the joint embedding space. The margin "m" is fixed for all terms across all training samples for the experiment and also across different loss components.

Motivation: The contributions of our work are as follows:

1) Augmented the image and text representation using ResNet and Word2Vec Models
2) Designed our own network to extract the joint embedding using transfer learning from classification.
3) Implemented loss function with two aspects bi-directional ranking and structure-preserving constraints.

4) Evaluate the impact of changing 'margin' and 'weight' values on the loss components.

As margin is the primary measure to define the difference/space between how similar and dissimilar images are in the embedding space, having an adaptive variable margin has the prospect of improving the embedding structure.

II. Related work

Learned embeddings from intermediate layers of powerful CNN classification models like VGG16 [14] and ResNet [15] have proved their ability to provide a base representation for efficient transfer learning. Such representations can be used in multiple domains such as medical image analysis [16], image super resolution [17] and indoor localization [18]. Learning such low dimensional features from scratch for each domain has proved to be inefficient, as the low-level features of images across domains highly correlate internally.

Word embedding models map the one-hot vector space to a continuous vector space in a much lower dimension than the conventional bag-of-words model. Word embeddings can be used for short text sentiment classification stating that the fixed number of dimensions in the word embedding model can facilitate more efficient computations [19]. In some studies, word embeddings are used in combination with other feature vectors for improved performance. The authors of [20] demonstrate that using word embeddings with classification models makes it possible to learn high dimensional word vectors practically and can be used to represent a large amount of data precisely. Word embedding features have since been applied to more and more text classification tasks [21].

One of the variations of Canonical Correlation Analysis, Kernel CCA, works on the principle of maximizing correlated non-linear projections extracted from multiple representations [12]. As an alternative to such correlation projections, a margin-based ranking loss technique was applied in DeVISE to learn linear transformations of visual and text features into latent shared space [22].

FaceNet [23] has shown that the method of mining triplets of matching and non-matching pairs during training significantly improves the representational efficiency of the embedding based on the similarity distance. [13] extended their work using two-branch neural nets. The experiments in [13] have shown that the structure-preserving constraints applied to learn embedding space across two different modalities achieved state-of-the-art results.

III. Problem definition

In this work, we propose to learn an image-text embedding representation using a two-view neural network with two layers of non-linearities on top of any representations of the image and text views. These representations were given by the outputs of two individual networks designed using transfer learning. Therefore, the presented work tries to map the classification approach(Figure 1) to a common embedding approach(Figure 2).

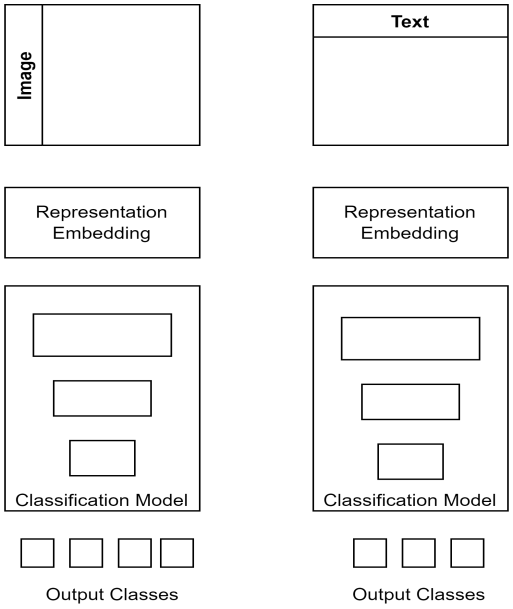


Fig. 1: Image and Text Classification

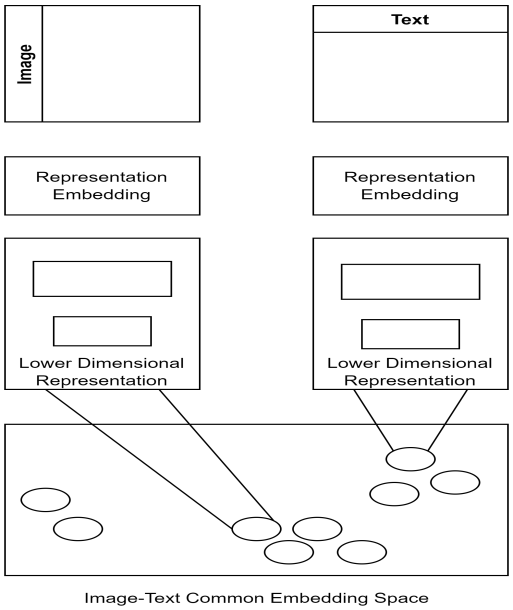


Fig. 2: Image-Text Combined Embedding

To train the branch network, a bi-directional loss is designed similar to the one proposed in the paper [24]. To this loss function, additional components were added to preserve the structure. Specifically, in the learned latent space, we want images and captions with similar meanings to be close to each other. The margin and tuning parameters (Lambda 1 λ_1, Lambda 2 λ_2, Lambda 3 λ_3) of different loss components in the bidirectional loss are varied to study the impact on the joint embedding. The loss function is further explained in section V.

IV. Image and Text Model Frameworks

The proposed model consists of two branches, one for images and the other for text. Each branch consists of a neural network with trainable and non-trainable layers. The individual branches for image and text were previously trained as classification models and sliced at the appropriate layers to join together to form the two-branch neural network. The following subsections outline the details of each of the individual neural network models.

A. The Image Model Branch

In image-sentence retrieval experiments, to represent images, the implementation details in [13] have been followed. ResNet 50 [15] was used instead of the 19-layer VGG model for extracting the image embeddings. Even though the Resnet network is much deeper than VGG, the training time is substantially lesser due to the usage of global average pooling layers instead of fully connected layers and reduces the complexity [15].

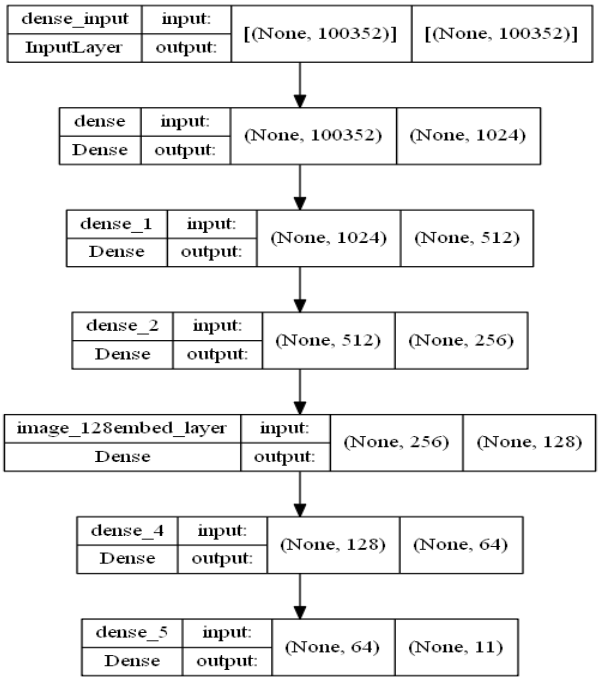


Fig. 3: Image Classification Model

The ResNet 50 model is loaded without the top classification layer to extract only the bottleneck features from the network. Following standard procedure, the original images are reduced to size 224 x 224 and have three colour channels. The output layer of the pre-trained model has a volume of 7 x 7 x 2048, which are the height, width, and the number of filters, respectively. The output is treated as a feature vector, and the list is flattened. Transfer learning is applied via feature extraction of the images. A simple feed-forward neural network architecture is trained on the feature vector to perform classification tasks based on the categories selected for the experiment, as shown in figure 3. The network is trained using an Adam optimizer with the learning rate parameter fixed at 0.01. The encoding of the length 128 is extracted from the built network.

B. The Text Model Branch

The second branch of the model framework includes a text classification model that is previously trained and then sliced to perform transfer learning. The following stages were involved in setting up and training the classification model. The five annotations(captions) of each image were collected and considered as the corpus for this model. The extracted text was tokenized, stemmed, lemmatized and transformed into a numeric representation with padding to ensure fixed-length sentences. The proposed classification model is a neural network constructed of deeply connected layers and LSTM(Long Short Term Memory) layers, as shown in figure 4.

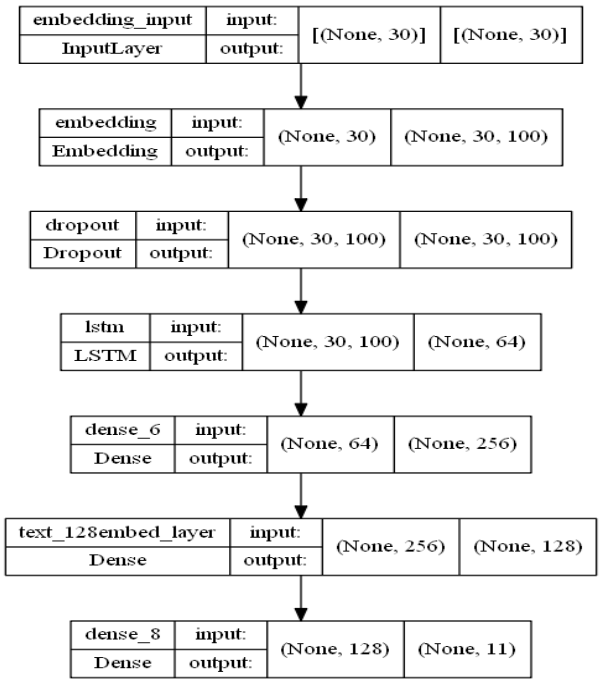


Fig. 4: Text Classification Model

The first layer of the model was initialized with an embedding matrix with 100-dimensional GloVe embeddings of all unique tokens present in the corpus. The model was trained with the Adam optimizer to perform the classification of texts into the categories selected for the experiment. After completion of training, the second last layer of the model was extracted to generate 128-dimensional embeddings of the text data passed through it. These embeddings are then mapped onto the joint embedding space in the next stage, along with the image embeddings that are generated parallelly.

V. The Two-Branch Model

In order to correlate the image and text embeddings internally, the computational graphs from both branches need to be aggregated together in a single Directed Acyclic Graph(DAG). Apart from that, a non-linear activation is required in the graph

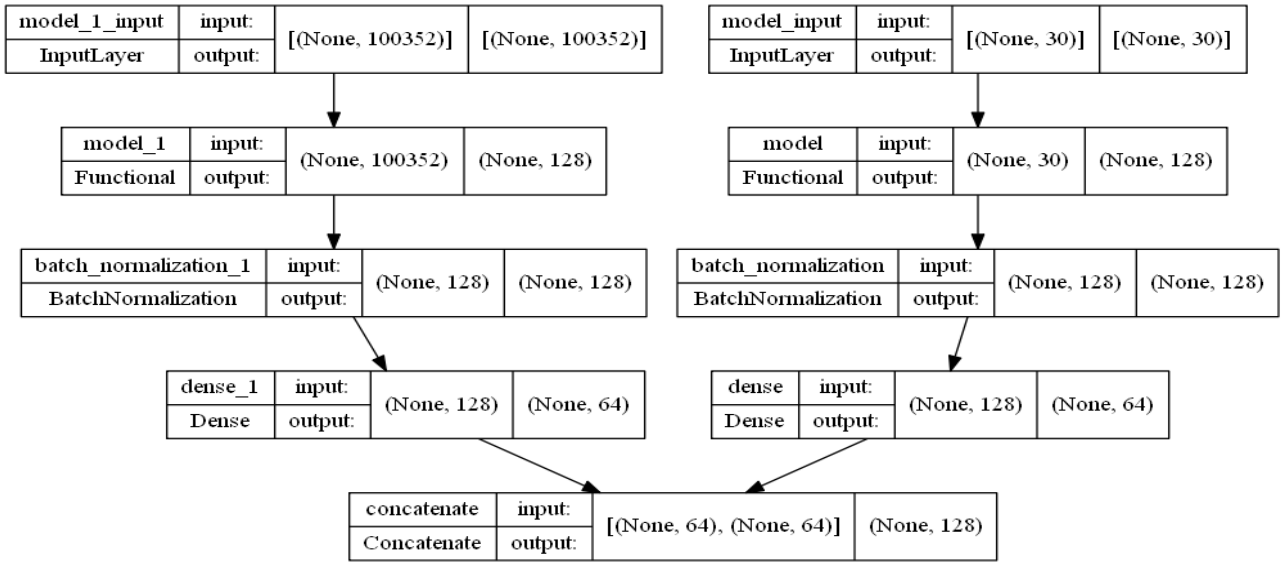


Fig. 5: Image-Text Combined Embedding Model

to model the non-linear behaviour of the data. Hence, a 64-dimensional Dense layer is added to both the branches, which are then concatenated together to form a 128-dimensional output of the combined model. Such concatenation allows the model to align multiple branches together and thus provides the ability to complete the backward propagation loop using a custom loss function. As seen in figure 5, the 128-dimensional outputs of both branches are passed through a Batch Normalization layer, a 64-dimensional dense layer, and are combined to get aggregated 128-dimensional representation of Image and Text together.

A. Dataset

In order to create a combined representation of images and text, the COCO dataset [25] was used for training, validation and testing loops of the model. It consists of images and five different annotations(captions) describing each image. For simplicity, the images were filtered from the dataset based on their dominance in the visual space, i.e. the images with a higher percentage of the primary category were chosen for the representational learning. In addition to that, a subset of eleven categories was chosen for the model as follows - person, car, truck, traffic light, dog, horse, zebra, bench, elephant, handbag and umbrella. Finally, any given instance of an image belongs to one category alone, which boosts the learning objective due to its image localization.

To train and assess the base networks for image and caption data, the initial dataset of size 6200 images was divided into the train, test, and validation datasets. For the combined two-branch network, the dataset was expanded to map every image to each of its five annotations(captions). After mapping, the dataset with 6200 instances is transformed into 31000 data points. For every pair of images and text, 128-dimensional features of the image and text were extracted.

Batches were created by shuffling the dataset and then sampling a fixed number of data points from it. As we calculate the loss by mining positive and negative instances for every data point, it is essential to ensure that every batch created includes at least two data points from the same category. In the case where a batch contains only one instance from a category, a random data point is swapped out from the most represented category for the required category. The training time of the combined model can be reduced by the integration of such a custom batch creation approach.

B. Margin-based triplet mining

Given training batch B, which consists of 32 pairs of (X, y), the triplet matching approach utilizes the distance metrics to create pairs of hard negatives and hard positive data points(Image and Text mapping), which can be used for generating a bi-directional ranking loss. Here, X represents a batch of data points which are a combination of Image Representation v_i of Image I and Text Representation ${t_i}^+$ of one of the matching annotations(captions) C_i for the image, and y represents the category of image and the matching text. For each anchor(representing text or image) of the dataset, the hardest negative and positive pairs were selected among the mini-batch. Such pairs are calculated based on the output category provided using y. This produces 32 triplets which are mined after applying the Euclidean distance metric. For instance, the 32 pairs of image and text embeddings of dimensions(64 dimensions each) are passed, and the distance is evaluated internally.

$$||v_i - t_j||^2 = ||v_i||^2 - 2 < v_i, t_j > + ||t_j||^2 \tag{1}$$

C. Training objective

The stochastic margin-based loss function is designed with two aspects, bi-directional ranking and structure-preserving

constraints [13] as shown below.

$$L(X, Y) = crossView + withinView \quad (2)$$

Cross-View Constraint: Cross-view preserves the bi-directional ranking loss. Given a training image v_i, the distance of the image is measured from the matching text ${t_j}^+$ and non-matching text ${t_j}^-$. Similarly, given a text t_i, matching image ${v_j}^+$, and non-matching image ${v_j}^-$ are measured. In addition to that, margin m enforces that the matching pairs are closer to each other as compared to non-matching pairs by some constraint m.

$$crossView = \sum max[0, m + d(x_i, y_j) - d(x_i, y_k)] + \lambda_1 \sum max[0, m + d(x'_i y'_j) - d(x'_i, y'_k)] \quad (3)$$

where,
i = image, j = positive sentence match,
k = negative sentence match, i' = sentence,
j' = positive image match, k' = negative image match

Within-View Constraint: Within-view constraint preserves the structure between neighbouring points of similar type(image or text); the images sharing the same meaning will be closer analogously. For instance, the distance between image pair x_i and x_j, which belong to the same category, should be less than image pair x_i and x_k, which do not belong to the same category. The same applies to sentences as well; that is, the distance between sentence pair y_i and y_j, which belong to the same category, should be less than sentence pair y_i and y_k which do not belong to the same category. Finally, the margin m enforces the matching points inside a boundary space while keeping the non-matching points outside the boundary space.

$$withinView = \lambda_2 \sum max[0, m + d(x_i, x_j) - d(x_i, x_k)] + \lambda_3 \sum max[0, m + d(y'_i, y'_j) - d(y'_i, y'_k)] \quad (4)$$

where,
i = image, j = positive sentence match,
k = negative sentence match, i' = sentence,
j' = positive image match,k' = negative image match

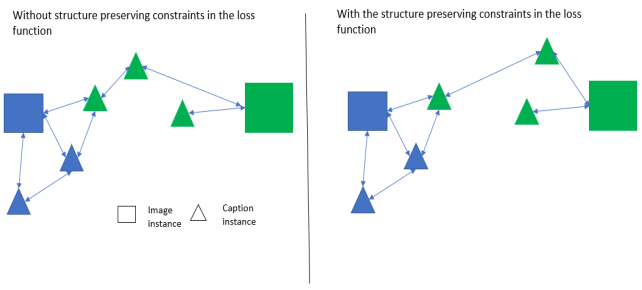


Fig. 6: Structure Preserving Constraints

The importance of margin is multi-fold as follows:

1) Responsible for pushing data points of a similar category but the different type(Image-to-Text) together.
2) Responsible for pushing data points of a different category and different types (Image-to-Text) further apart.
3) Enforces distance between data points of similar category and same type(Image-Image and Text-Text) closer.
4) Enforces distance between data points of a different category and same type(Image-Image and Text-Text) to be further apart.

The loss gradients of these components over the triplets are back-propagated through the network, and the parameter weights are updated to achieve a well-separated region which is used later used for querying utilities.
The lambda values(λ_1, λ_2 and λ_3) are added to control the contribution of these loss terms. These weights regularize the loss components and provide a dynamic learning objective to the model. λ_1 regularises the ranking loss in the cross-view component, while λ_2 and λ_3 manage the weightage of within-view terms in the overall loss value.

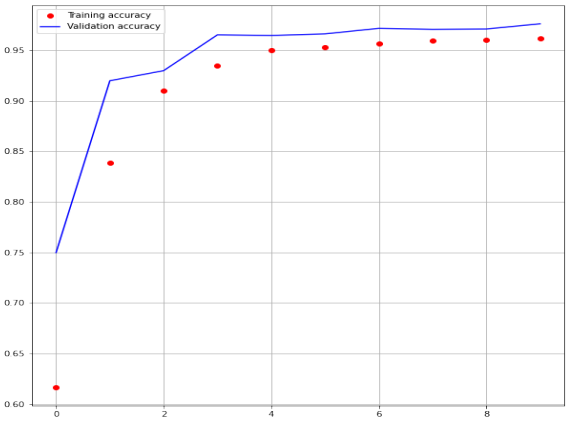
The training objective is to minimize a triplet loss function, which applies a margin-based penalty to balance the cross-view and within view losses. A margin-based penalty is applied to an incorrect annotation when it gets ranked higher than a correct one for describing an image as well as ensuring that for each annotation, the corresponding image gets ranked higher than the unrelated ones.

VI. Experiments

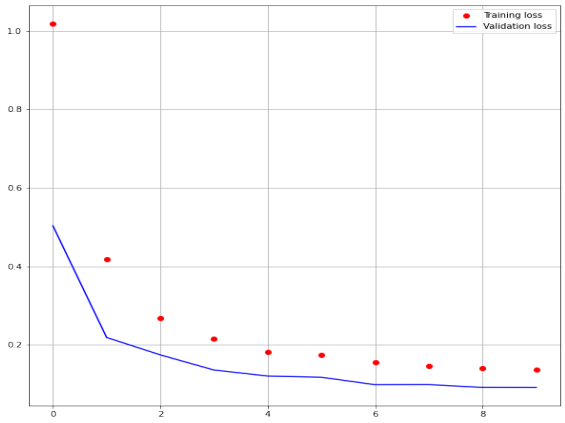
In this section, we discuss the experiments performed for multiple components of the system for the task of Image Retrieval using Text on MSCOCO dataset [25].

A. Network Branch Settings

In this work, we used the ResNet50 model's final internal activations for representing an image in a lower-dimensional space. Following such transformation, the image representation dataset is used to train the Image model for three epochs. During this training phase, the training accuracy increased to 98% from 80%, and the validation accuracy increased to 92%. In terms of loss, in the categorical cross-entropy, the training loss reduced from 0.63 to 0.04 and validation loss reduced from 0.30 to 0.27. Also, we used Word2Vec's Glove Embedding trained on 6 Billion parameters and 400K vocabulary terms with 100 dimensions as output. In addition to this, we used 30 word as our base length, where the remaining words for the smaller length texts would be padded. During the training phase of this branch, we trained the model for ten epochs, and the training loss reduced from 1.01 to 0.13, and training accuracy increased from 61% to 96%. In-depth training curves along with validation, losses are shown in figure 7



(a) Training vs Validation Accuracy (Accuracy vs Iterations)



(b) Training vs Validation Loss (Loss vs Iterations)

Fig. 7: Performance Metrics

B. *Tuning Margin and Weight for the loss function*

m	m_1	m_2	m_3	λ_1	λ_2	λ_3	**Average Loss of last 10 iterations**
0.2	0.15	0.1	0.2	2	1	0.5	0.622261
0.3	0.15	0.1	0.2	2	1	0.5	0.713236
0.1	0.5	0.1	0.2	2	1	0.5	1.207527
0.1	0.75	0.1	0.2	2	1	0.5	1.712266
0.1	0.15	0.2	0.2	2	1	0.5	0.508106
0.1	0.15	0.3	0.2	2	1	0.5	0.5138
0.1	0.15	0.1	0.5	2	1	0.5	0.663202
0.1	0.15	0.1	0.2	1	1	0.5	**0.356512**
0.1	0.15	0.1	0.2	3	1	0.5	0.660068
0.1	0.15	0.1	0.2	2	0.5	0.5	0.517828
0.1	0.15	0.1	0.2	2	0.25	0.5	0.506008
0.1	0.1	0.1	0.15	2	1.5	0.2	**0.37663**
0.1	0.1	0.1	0.1	2	1.5	0.5	**0.382047**
0.1	0.15	0.1	0.2	2	1	0	0.472505
0.1	0.15	0.1	0.2	2	1	0.5	0.505144
0.5	0.15	0.1	0.2	2	1	0.5	0.907083
0.9	0.15	0.1	0.2	2	1	0.5	1.312319
0.1	0.15	1	0.2	2	1	0.5	0.535979
0.1	0.15	0.1	1	2	1	0.5	0.908248

TABLE I: Average loss based on Hyper-Parameters

In order to best optimize the Margin value and the Weight parameters, we experimented with different combinations, and found that the combination of 0.1, 0.15, 0.1, 0.2, 1, 1 and 0.5 as margin m, margin m_1, margin m_2 margin m_3, weight λ_1, weight λ_2 and weight λ_3. Such a combination was able to reach the optimum loss value in quickest time, and reached lowest value of loss 0.356. Such faster training curve can be seen in figure 8

C. *Image Retrieval using Text*

We train our combined network using Adam optimiser with the learning rate of 0.0001, $beta_1$ = 0.9, $beta_2$ as 0.999 and $epsilon$ as e^{-7}. To accelerate the convergence during the training phase and to also make gradient updates more stable, we apply batch normalization [26].

During inference, the text and image encoders can yield image and textual query feature vectors, which are part of the common embedding space of size 128. These feature vectors can be used to search for images and queries with similar visual or textual properties. The similarity metric used for matching the query and MSCOCO dataset is the same as in the training phase. Thus, the Euclidean distance is used to compare and rank search results for the query. The catalog image and textual features can be pre-computed offline and stored.

The model was tested on multiple images and search queries to track the recall, which is the percentage of queries for which at least one correct ground truth was ranked among the top K (5) matches. Even with the small number of data points and limited compute capabilities, the model performed well on K(5); on average, 1-2 images matched with the search text out of the top 5 results, bringing the successful search from 31000 data points with around 30% accuracy. Some matching and non-matching examples of image search using text are shown in figure 9.

VII. Conclusion

This work focused on creating a common image-text embedding using a two-branched neural network. The research also explored the possibility of multiple combinations of Weight and Margin parameters. The model was trained by balancing the bi-directional ranking components and the structure-preserving components. Finally, our work demonstrated the faster and more efficient learning capabilities of the model based on variable loss components. This work can be further extended to a larger dataset, using higher compute capabilities, and adaptive tuning can be performed on such hyperparameters.

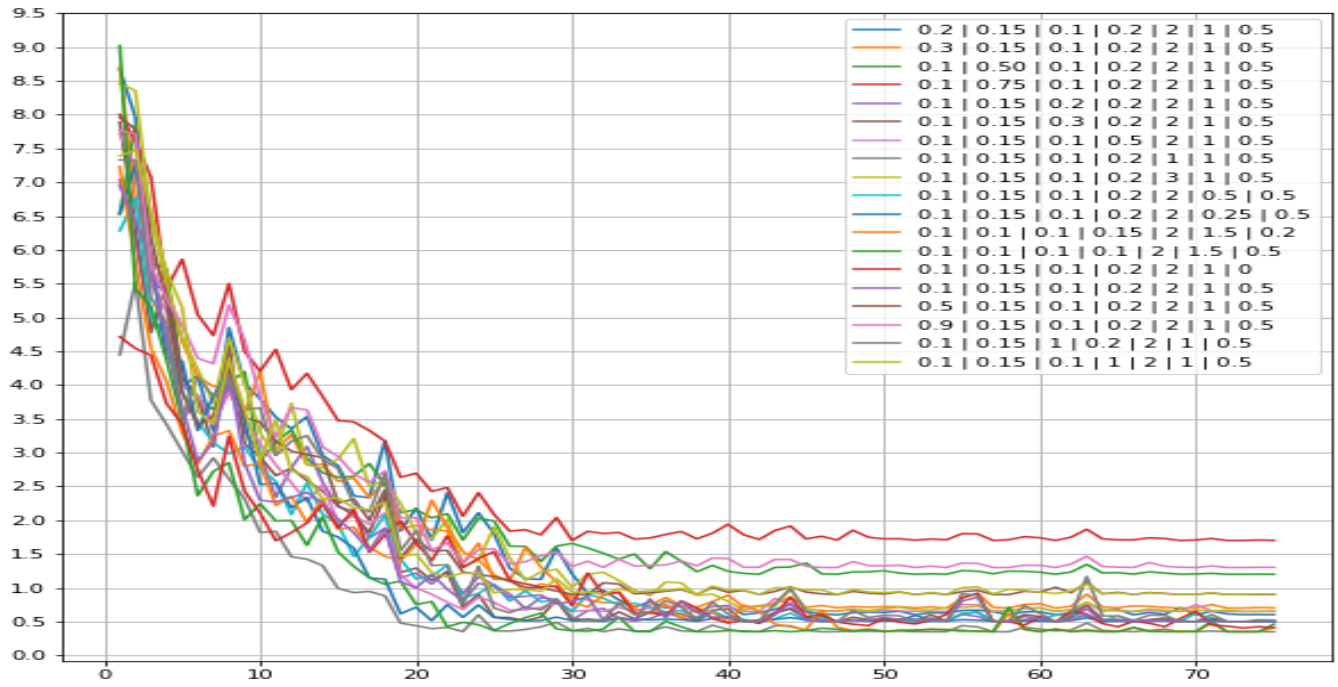
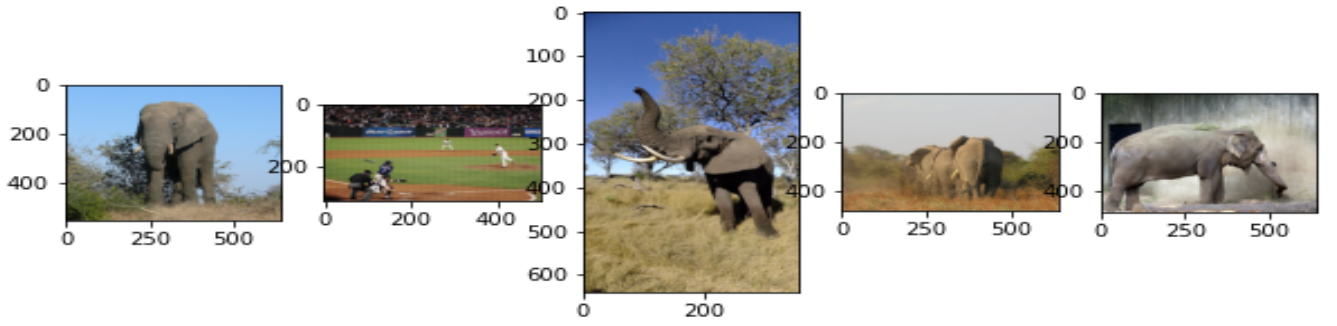
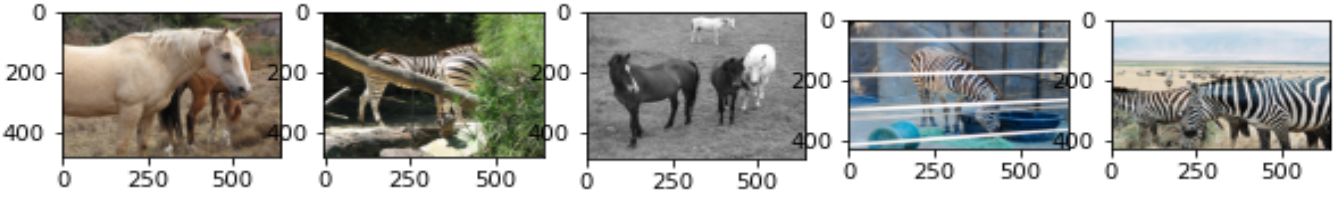


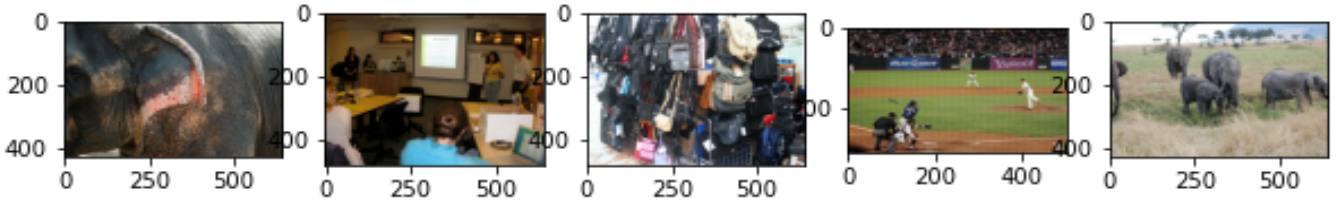
Fig. 8: Training curve vs Hyperparameters (Format: $m|m_1|m_2|m_3|\lambda_1|\lambda_2|\lambda_3$) (Loss vs Iterations)



(a) Search Text: ”walking elephant”



(b) Search Text:”zebra is eating”



(c) Search Text: ”A photograph studio”

Fig. 9: Image retrieval using Text

Acknowledgements: This research is carried out under the supervision of Prof. Dr. Jeff Orchard and under the curriculum of CS 679, Neural Networks course of the University of Waterloo.

REFERENCES

[1] W. Wang, X. Yang, B. C. Ooi, D. Zhang, and Y. Zhuang, "Effective deep learning-based multi-modal retrieval," *The VLDB Journal*, vol. 25, no. 1, pp. 79–101, 2016.

[2] K. Wang, Q. Yin, W. Wang, S. Wu, and L. Wang, "A comprehensive survey on cross-modal retrieval," *arXiv preprint arXiv:1607.06215*, 2016.

[3] K. Chen, T. Bui, C. Fang, Z. Wang, and R. Nevatia, "Amc: Attention guided multi-modal correlation learning for image search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2644–2652, 2017.

[4] P. Hu, L. Zhen, D. Peng, and P. Liu, "Scalable deep multimodal learning for cross-modal retrieval," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, (New York, NY, USA), p. 635–644, Association for Computing Machinery, 2019.

[5] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "Mdetr-modulated detection for end-to-end multi-modal understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1780–1790, 2021.

[6] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[7] W. Ying, Y. Zhang, J. Huang, and Q. Yang, "Transfer learning via learning to transfer," in *International Conference on Machine Learning*, pp. 5085–5094, PMLR, 2018.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[9] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717–1724, 2014.

[10] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[11] A. Tifrea, G. Bécigneul, and O.-E. Ganea, "Poincar\'e glove: Hyperbolic word embeddings," *arXiv preprint arXiv:1810.06546*, 2018.

[12] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[13] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5005–5013, 2016.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[16] M. Mateen, J. Wen, S. Song, and Z. Huang, "Fundus image classification using vgg-19 architecture with pca and svd," *Symmetry*, vol. 11, no. 1, p. 1, 2018.

[17] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3516–3525, 2020.

[18] I. Ha, H. Kim, S. Park, and H. Kim, "Image retrieval using bim and features from pretrained vgg network for indoor localization," *Building and Environment*, vol. 140, pp. 23–31, 2018.

[19] J.-H. Wang, T.-W. Liu, X. Luo, and L. Wang, "An lstm approach to short text sentiment classification with word embeddings," in *Proceedings of the 30th conference on computational linguistics and speech processing (ROCLING 2018)*, pp. 214–223, 2018.

[20] S. Li, J. Hu, Y. Cui, and J. Hu, "Deeppatent: patent classification with convolutional neural networks and word embedding," *Scientometrics*, vol. 117, no. 2, pp. 721–744, 2018.

[21] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014.

[22] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," *Advances in neural information processing systems*, vol. 26, 2013.

[23] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[24] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

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

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.