University of Insubria

Department of Theoretical and Applied Science (DiSTA)

PhD Thesis in Computer Science
XXXII Cycle of Study

Multimodal Representation and Learning

Candidate:
Shah Nawaz

Thesis Advisor:
Prof. Ignazio Gallo

October, 2019

To my Family

Contents

List of Figures

List of Tables

Acknowledgements

First, I would like to express my sincere gratitude to my advisor Prof. Ignazio Gallo and my colleague Alessandro Calefati. I would like to thank them for the continuous support of my doctoral study and research, for their patience, motivation, enthusiasm, and vast knowledge. Their guidance helped me in all the time of research and writing of this thesis.

I gratefully acknowledge Muhammad Kamran Janjua (National University of Sciences and Technology, Islamabad Pakistan), Muhammad Umer Anwaar (Technical University of Munich, Germany), Nisar Ahmed (University of Engineering and Technology, Lahore Pakistan) and Dr. Arif Mahmood (Information Technology University, Lahore Pakistan) for their collaboration in my research activities.

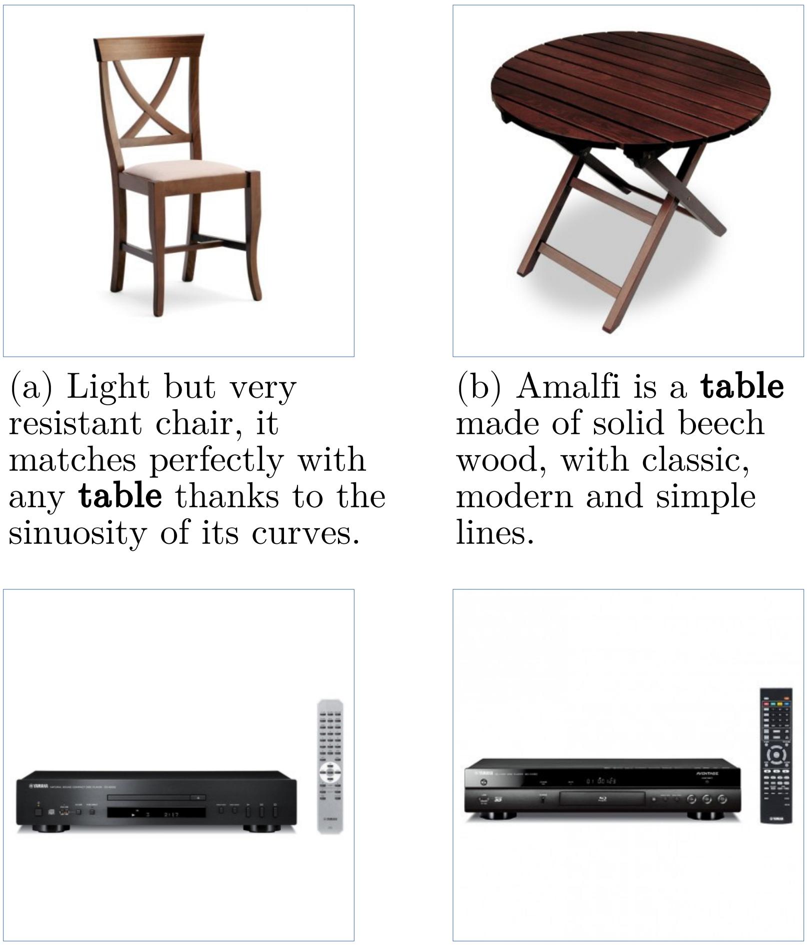
Furthermore, I acknowledge 7Pixel that made my PhD work possible providing ideas, data, funds and place for the research activities.

Last, but not least, I would like to thank my family for providing me with the needed support to succeed in this doctoral work.

1
Introduction

1.1 Motivation

Deep learning has remarkably improved the state-of-the-art speech recognition, visual object detection, object recognition and text processing tasks [1]. Majority of these techniques focused on unimodality (images, text, speech, etc.), however, real-world scenarios present data in a multimodal fashion – we see objects, listen sounds, feel texture, smell odours and taste flavours. Multimodality refers to the fact that the real-world concepts can be described by multiple modalities. Moreover, recent years has seen an explosion in multimodal data on the web. Typically, users combine text, image, audio or video to sell a product over an e-commence platform or express views on social media. It is well-known that multimodal data may provide enriched information to capture a particular “concept” than single modality [2]. For example, Figure 1.1 shows two adverts typically available on an e-commerce platform, where two visual objects have seemingly similar captions in the first row but dissimilar images. On the second row, we have two different captions but seemingly similar images in the second row. Typically, such scenarios are faced in multimodal classification. Similarly, Figure 1.2 shows two examples of multimodal data collected from a social media platform. If we consider only the text descriptions, entities may be wrongly labelled. Therefore, visual context is beneficial to resolve ambiguities. In addition, various modalities generally carry different kinds of information that may provide enrich understanding; for example, the visual signal of a flower may provide happiness; however, its scent might not be pleasant. Multimodal information may be useful to make an informed decision.



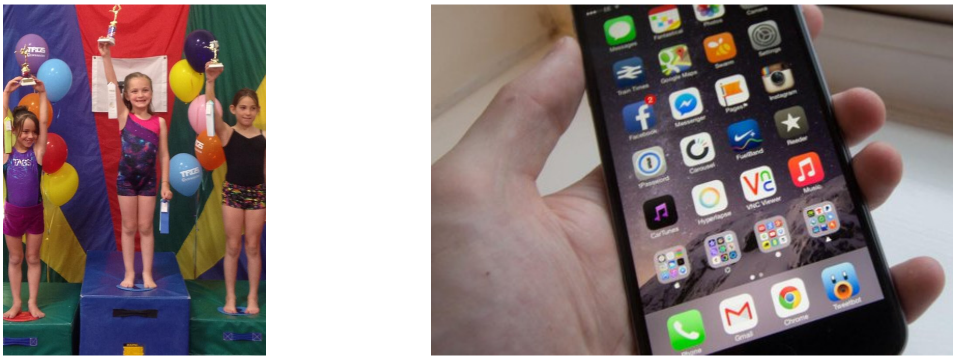
(a) Light but very resistant chair, it matches perfectly with any **table** thanks to the sinuosity of its curves.

(b) Amalfi is a **table** made of solid beech wood, with classic, modern and simple lines.

(c) Yamaha Cd-s300 Black **CD playe**r Black hifi. Supported audio formats: MP3, WMA. Formats: CD, CD-R RW.

(d) Yamaha BD-A1060 **Bluray player** Upscaling 4K 3D SACD USB Bluetooth Wi-Fi Aventage

Figure 1.1: In the top row, an example of ambiguous text descriptions that can be disambiguated with the analysis of the accompanying images. In the bottom row, examples of ambiguous images that can be disambiguated with the analysis of the associated text descriptions. Generally, e-commerce platforms have such ambiguous examples.



(a) My daughter got 1 place in [Apple valley **LOC**] Tags gymnastics.

(b) Apple **ORG**] 's latest [iOS **OTHERS**] update is bad for advertisers.

Figure 1.2: Two NER multimodal examples show how some entities in the text can be correctly tagged in combination with visual information. Looking only at the text, the word *Apple* is ambiguous in the text description on the left because it can be interpreted as *Location* (LOC) or as *Organization* (ORG).

It is therefore important to perform multimodal learning to understand the web and the world around us. We cannot afford to have a single model for every concept. However, it is challenging to interpret various modalities together because each modality has a different representation and correlational structure. For example, the text is typically represented as discrete sparse word count vectors whereas an image is represented using dense and real-value features. It is therefore difficult to capture cross-modal interactions between modalities than intra-modal interactions among the same modality. It is interesting to note that the input data from multiple modalities may contain structure however, it is challenging to discover the highly non-linear relationships that exist. Moreover, the data may contain noise along with missing values at the input. Therefore, meaningful representation should be extracted from multiple modalities to learn joint representations for multimodal applications. Srivastava and Salakhutdinov [3] identify the following desirable properties for good multimodal representations.

- It is challenging to obtain joint representation because multiple modalities data may be heterogeneous. It is therefore important that the joint representation should be meaningful and complement the corresponding “concepts” at the input level.
- The joint representation form various modalities provides more information than the representation from individual modalities however, modalities in real-world scenarios may be missing. It is therefore important that the model provides meaningful representation in such situations.

- The missing modalities in some case should be fill-in with the help of observed one.
- Typically, the extracted representation is employed in various applications including cross-modal verification and retrieval. It is therefore important that the extracted representation should be discriminative for improved performance.

In recent years, the combination of various modalities has been extensively studied to solve various tasks including classification [4, 5, 6], cross-modal retrieval [7] semantic relatedness [8, 9], Visual Question Answering [10, 11], image captioning [12, 13], multimodal named entity recognition [14, 15, 16, 17].

1.2 Challenges

This doctoral thesis identifies and explores the following core technical challenges concerning multimodal representation and learning. Besides, we focus on three modalities including text, audio and visual signals for various multimodal applications such as classification, cross-modal retrieval and verification on benchmark datasets.

- The first fundamental challenge is learning how to represent and summarize multimodal data to exploit meaningful information from individual modality. However, the heterogeneity nature of multimodal data makes it difficult to construct joint representation. For example, language is often symbolic, while audio and visual modalities are represented as signals.
- The second challenge is to join information from multiple modalities to perform various tasks. For example, for audio-visual speech verification, the visual description of the face is fused with the audio signal to verify if audio and face image belongs to the same identity or not. The information coming from multiples modalities may have different predictive power, with possibly missing information in at least one of the modalities. In other words, the joint representation should have discriminative power to be employed for various tasks.

The unimodal representations have been extensively studied [18, 19]. In recent years, there is a paradigm shift in unimodal representations from hand-crafted for particular applications to data-driven representations. For example, visual descriptors learned from data using neural architectures such as Convolutional Neural Networks (CNNs) have outperformed hand-craft descriptors, namely the scale invariant feature transform [20]. Therefore, representations will be extracted from CNNs in accordance with guidelines from Bengio et al. [18] for unimodal representations and Srivastava and Salakhutdinov [3] for multimodal representations.

1.3 Contributions

In a multimodal approach, typically, data is obtained from various modalities and representation for a particular modality is extracted. In this doctoral thesis, representations from individual modalities are improved to extract enhanced representations. These contributions are listed as follow:

- In Chapter 2, "**visual word embedding scheme**" is presented to transform word embedding to visual space. The aim of the approach is to employ state-of-the-art image classification models for text classification. The visual word embedding is evaluated against state-of-the-art text classification methods on 8 benchmark datasets. The work results in the following publications:

  1. Shah Nawaz, Alessandro Calefati, Ignazio Gallo. Visual Word Embedding for Text Classification. Submitted: Journal of Machine Vision and Applications 2019.
  2. Ignazio Gallo, Shah Nawaz, Alessandro Calefati. Semantic Text Encoding for Text Classification using Convolutional Neural Networks. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 5, pp. 16-21). IEEE.

- In Chapter 3, multimodal framework named "**deep latent space representations**" is presented for cross-modal retrieval, matching and verification. The framework is coupled with a single stream network to bridge the gap between multiple modalities (text/image and audio/image) without needing a separate network for each modality. It is interesting to note that the text embedding in the multimodal framework is extracted from "**visual word embedding scheme**". In addition, the visual word embedding helped to employ single stream network because representations from text and the visual signal is extracted with the same neural network. The cross-modal retrieval application is evaluated against state-of-the-art methods on a benchmark dataset named MSCOCO. Similarly, the cross-modal matching verification application is evaluated on VoxCeleb benchmark dataset that includes audio and visual information. This work results in the following publications covering cross-modal retrieval and verification applications evaluated aganist state-of-the-art works:

  1. Shah Nawaz, Muhammad Kamran Janjua, Ignazio Gallo, Arif Mehmood, Alessandro Calefati. Deep Latent Space Learning for Cross-modal Mapping of Audio and Visual Signals. In Digital Image Computing: Techniques and Applications 2019.

  2. Shah Nawaz, Muhammad Kamran Janjua, Alessandro Calefati, Ignazio Gallo and Arif Mehmood. Do Cross-Modal Systems Leverage Semantic Relationships? In International Conference on Computer Vision Workshop 2019.

- In Chapter 4, "**deep fused representations framework**" is presented for multimodal classification. The multimodal representation is obtained by fusing "**visual word embedding**" and visual signals. Finally, the information enriched (fused) image is classified with single stream state-of-the-art image classification model. The proposed approach is evaluated against state-of-the-art methods on three benchmark multimodal datasets including Ferramenta, UPMC Food 101 and Amazon Product Data. This work results in the following publication:

  1. Shah Nawaz, Alessandro Calefati, Muhammad Kamran Janjua, Muhammad Umer Anwaar, Ignazio Gallo. Learning Fused Representations for Large Scale Multimodal Classification. In IEEE Sensors Letters 2018.

- In Chapter 5, an "**inwardly scale feature representations**" in proportion to projecting them onto a hypersphere manifold for discriminative analysis is presented. The proposed approach will render similar instances closer while dissimilar instances distant. For this purpose, an inward scaling layer is paired with different deep network architectures. Extensive experiments are performed on multitude of datasets to establish the empirical gain achieved with the purposed method on image classification and retrieval. Comparison with current state-of-the-art techniques demonstrated the excellent performance of the purposed method.
- In Chapter 6, a new handwritten dataset named "**Urdu-Characters**" with set of classes suitable for deep metric learning is created. The performance of two state-of-the-art deep metric learning methods i.e. Siamese and Triplet network, is compared. We show that a Triplet network is more powerful than a Siamese network. In addition, we show that the performance of a Triplet or Siamese network can be improved using the most powerful underlying Convolutional Neural Network architectures. This work results in the following publication:

  1. Shah Nawaz, Alessandro Calefati, Nisar Ahmed Rana, Ignazio Gallo. Hand Written Characters Recognition via Deep Metric Learning. In 13th IAPR International Workshop on Document Analysis Systems (DAS), pp. 417-422. IEEE, 2018.

In Chapter 2, 5 and 6, representations from individual modalities are discussed which in turn are employed in two multimodal frameworks for classification, cross-modal retrieval and verification in Chapter 3 and 4.

2

Visual Word Embedding for Text

2.1 Introduction

Text classification is a common task in Natural Language Processing. Its goal is to assign a label to a text document from a predefined set of classes. In recent years, CNNs have remarkably improved performance in image classification [21, 22, 23] and researchers have successfully transferred this success into text classification [24, 25]. Image classification models [21, 23] are adapted to accommodate text [26, 24, 25]. We, therefore, leverage on the recent success in image classification and present a novel text classification approach to cast text documents into visual domain to categorize text with image classification models. Our approach transforms text documents into encoded images or visual embedding capitalizing on Word2Vec word embedding. Word embedding models [27, 28, 29] for text classification convert words into vectors of real numbers. Typically word embedding models are trained on a large corpus of text documents to capture semantic relationships among words. Thus, these models can produce similar word embeddings for words occurring in similar contexts. We exploit this fundamental property of word embedding models to transform a text document into a sequence of colors (visual embedding), obtaining an encoded image, as shown in Figure 2.1. Intuitively, semantically related words obtain similar colors or encodings in the encoded image while uncorrelated words are represented with different colors. Interestingly, these visual embeddings are recognized with state-of-the-art image classification models.

We present a text classification approach to transform word embedding of text documents into the visual domain. We evaluated the method on several large scale datasets

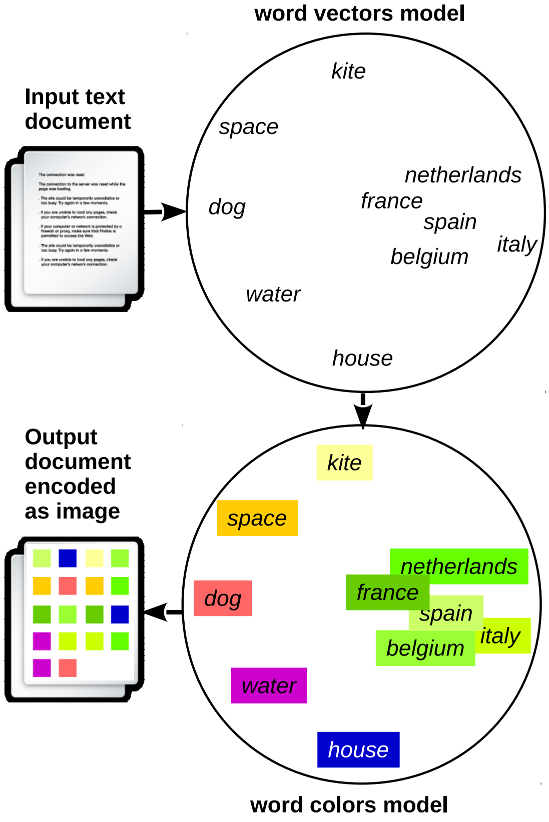


Figure 2.1: We exploited a well-known property of word embedding models: semantically correlated words obtain similar numerical representation. It turns out that if we interpret real-valued vectors as a set of colors, it is easy for a visual system to cope with relationships between words of a text document. It can be observed that green colored words are related to countries, while other words are represented with different colors.

obtaining promising and, in some cases, state-of-the-art results.

2.2 Related Work

Deep learning methods for text documents involved learning word vector representations through neural language models [27, 28]. These vector representations serve as a foundation of our work where word vectors are transformed into a sequence of colors or visual embedding. Image classification model is trained and tested on these visual embeddings. Kim [26] proposed a simple shallow neural network with one convolution layer followed by a max pooling layer over time. The final classification is performed with one fully connected layer with drop-out. The work in [24] presented rather deep convolutional neural network for text classification. The network is similar to the convolutional network in computer vision [21]. Similarly, Conneau et al. [25] presented a deep architecture that operates at character level with 29 convolutional layers to learn hierarchical representations of text. The architecture is inspired by recent progress in computer vision [30, 31]. In our work, we leverage on recent success in computer vision, but instead of adapting the deep neural network to be fed with raw text information, we propose an approach that transforms word embedding of text documents into encoded text. Once we have encoded text, we can apply state-of-the-art deep neural architectures used for image classification. We compared our proposed approach with deep learning models based on word embedding and lookup tables along with the method proposed in [24] and [25, 32] using the same datasets. In Section 2.5, experimental results of our proposed approach are shown, highlighting that, in some cases, it overtakes state-of-the-art results while in other cases, it obtains comparable results.

2.3 Proposed Approach

In this section, we present our approach to transforming Word2Vec word embedding into the visual domain. In addition, we explained the understanding of CNNs with the propose approach.

2.3.1 Encoding Scheme

The proposed encoding approach is based on Word2Vec word embedding [27]. We encode a word t_k belonging to a document D_i into an artificial image of size $W \times H$. The approach uses a dictionary $F(t_k, v_k)$ with each word t_k associated with a feature vector $v_k(t_k)$ obtained from a trained version of Word2Vec word embedding model. Given the word t_k, we obtained a visual word $\hat{t}_k$ having width V that contains a subset of a feature vector, called superpixels. A superpixel is a square area of size $P \times P$ pixels with a uniform color that represents a sequence of contiguos features $(v_{k,j}, v_{k,j+1}, v_{k,j+2})$ extracted as a sub-vector of v_k. A graphical representation is shown in Fig. **??**. We normalize each component $v_{k,j}$ to assume values in the interval $[0 \ldots 255]$ with respect to k, then we interpret triplets from feature vector v_k as RGB sequence. For this very reason, we use feature vector with a length multiple of 3.

The blank space s around each visual word $\hat{t}_k$ plays an vital role in the encoding approach. We found out that the parameter s is directly related to the shape of a visual word. For example, if $V = 16$ pixels, then s must also have a value close to 16 pixels to let the network understand where a word ends and another begins.

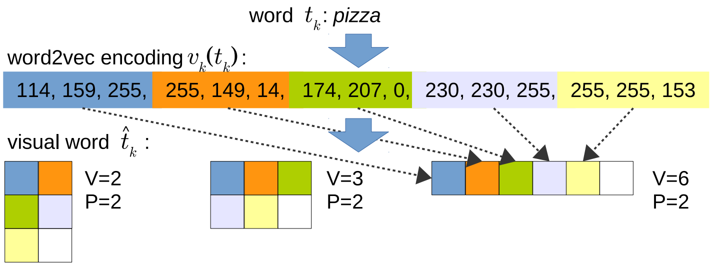


Figure 2.2: In this example, the word "*pizza*" is encoded into a visual word $\hat{t}_k$ based on Word2Vec feature vector with length 15. This visual word can be transformed into different shapes, varying the V parameter (in this example $V = 2, 3, 6$ superpixels).

2.3.2 Encoding Scheme with CNN

It is well understood that a CNN can learn to detect edges from image pixels in the first layer, then use the edges to detect trivial shapes in the next layer, and then use these shapes to infer more complex shapes and objects in higher layers [33]. Similarly, a CNN trained on our proposed visual embedding may extract features from various convolutional layers (see example in Fig. 2.4). We observed that the first convolutional layer recognizes some specific features of visual words associated to single or multiple superpixels. The remaining CNN layers aggregate these simple activations to create increasingly complex relationships between words or parts of a sentence in a text document.

To numerically illustrate this concept, we use the receptive field of a CNN. The receptive field r is defined as the region in the input space that a particular CNN feature is looking at. For a convolution layer of a CNN, the size r of its receptive field can be computed by the following formula:

$$r_{out} = r_{in} + (k - 1) \cdot j_{in} \tag{2.1}$$

where k is the convolution kernel size and j is the distance between two consecutive features. We can compute the size of the receptive field of each convolution layer using the formula in Eq. 2.1. For example, the five receptive field of an AlexNet, showed in Fig. 2.3, have the following sizes: *conv1* 11 × 11, *conv2* 51 × 51, *conv3* 99 × 99, *conv4* 131 × 131 and *con5* 163 × 163. This means that the *conv1* of an AlexNet, recognizes a small subset of features represented by superpixels, while the *conv2* can recognize a visual word (depending on the configuration used for the encoding), up to the *con5* layer where a particular feature can simultaneously analyze all the visual words available in the input image.

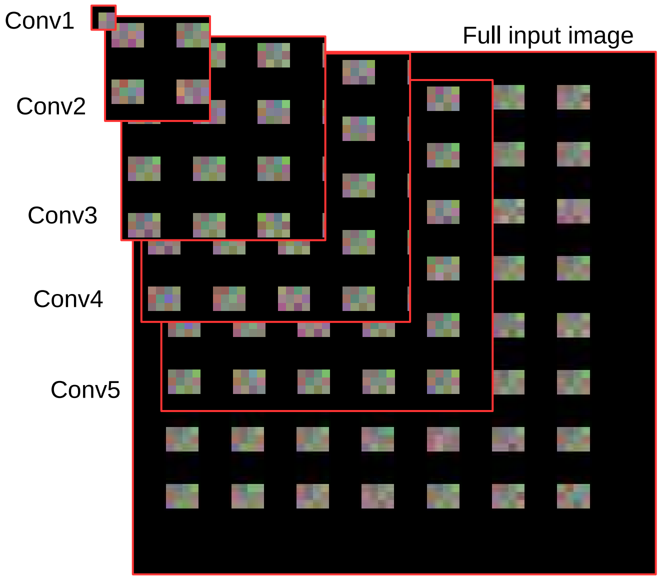


Figure 2.3: The receptive fields of the five convolution layers of an AlexNet. Each receptive field is cut from a 256 × 256 image to analyze the quantity of visual words that each *conv* layer is able to analyze on each pixel of its feature map.

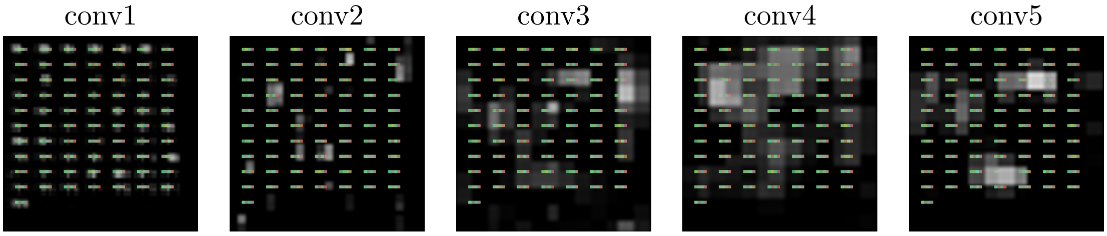


Figure 2.4: An example of five feature maps (conv1,. . . , conv5) displayed over the input image of the DBPedia dataset. In this particular example images are encoded with 12 Word2Vec features and 16 pixels of space between visual words. The convolutional map generated by the conv1 layer shows activations of individual superpixel or sequence of superpixels, while other convolutional layers show larger activation areas affecting more visual words.

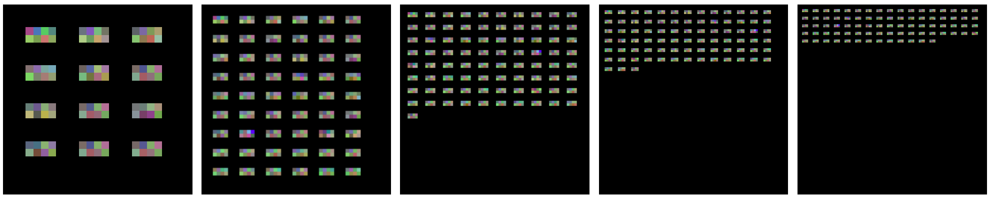


Figure 2.5: Five different sizes of encoded image (100×100, 200×200, 300×300, 400×400, 500 × 500) obtained using the same document belonging to the 20news-bydate dataset. All images use the same encoding with 24 Word2Vec features, space $s = 12$, superpixel size 4 × 4. It is important to note that the two leftmost images cannot represent all words in the document due to the small size.

2.4 Datasets

Zhang *et al.* [24] introduced several large-scale datasets which covers several text classification tasks such as *sentiment analysis, topic classification* or *news categorization.* In these datasets, the number of training samples varies from several thousand to millions, which is considered ideal for deep learning based methods. In addition, we used the 20news-bydate dataset to test various parameters associated with the encoding approach. A summary of the statistics for each dataset is listed in Table 2.1.

Table 2.1: Statistics of 20news-bydate and large-scale datasets presented in Zhang *et al.* [24]. The 2 rightmost columns show the average (Avg) and standard deviation (Std) for the number of words contained in text documents.

Dataset	**Classes**	**Training**	**Test**	**Avg**	**Std**
20news-bydate	4	7,977	7,321	339	853
AG's News	4	120,000	7,600	43	13
Sogou News	5	450,000	60,000	47	73
DBPedia	14	560,000	70,000	54	25
Yelp Review Polarity	2	560,000	38,000	138	128
Yelp Review Full	5	650,000	50,000	140	127
Yahoo! Answers	10	1,400,000	60,000	97	105
Amazon Review Full	5	3,000,000	650,000	91	49
Amazon Review Polarity	2	3,600,000	400,000	90	49

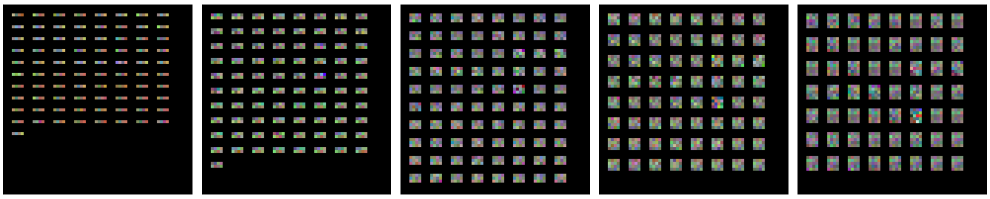


Figure 2.6: Five encoded images obtained using different Word2Vec features length and using the same document belonging to the 20news-bydate dataset. All the images are encoded using space $s = 12$, superpixel size 4×4, image size $= 256 \times 256$ and visual word width $V = 16$. The two leftmost images contain all words in the document encoded with 12 and 24 Word2Vec features respectively, while 3 rightmost encoded images with 36, 48 and 60 features length cannot encode entire documents.

2.5 Experiments and Results

The aim of these experiments is as follow: (i) to evaluate configuration parameters associated with the encoding approach; (ii) to compare the proposed approach with other deep learning methods.

In experiments, percentage error is used to measure the classification performance. The encoding approach mentioned in Section 2.3.1 produces encoded image based on Word2Vec word embedding. These encoded images can be used to train and test a CNN. We used AlexNet [21] and Googlenet [22] architectures as base models from scratch. We used a publicly available Word2Vec word embedding with default configuration parameters as in [27] to train word vectors on all datasets. Normally, Word2Vec is trained on a large corpus and used in different contexts. However, in our work, we trained this model with the same training set for each dataset.

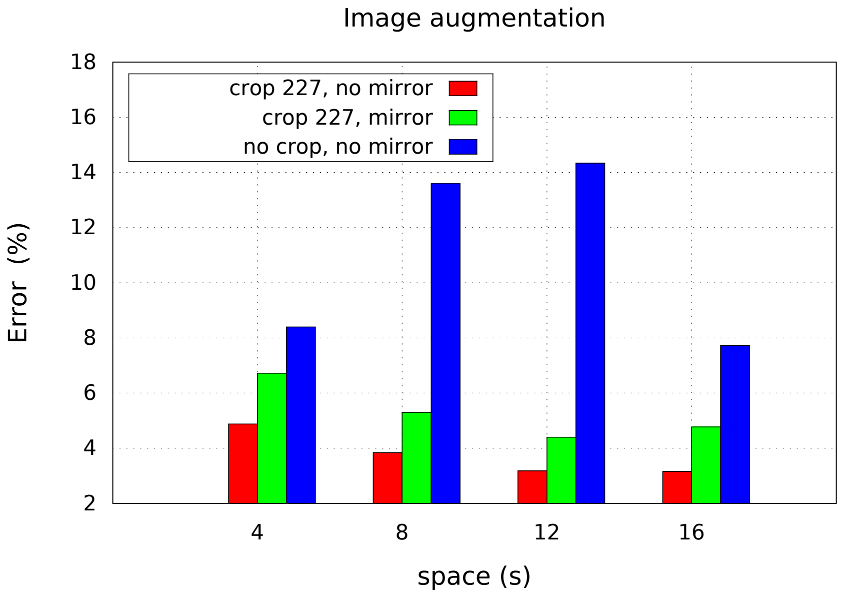
2.5.1 Parameters Setting

We used 20news-bydate dataset to perform a series of experiments with various settings to find out the best configuration for the encoding scheme.

In our first experiment, we changed the space s among visual words and Word2Vec feature length to identify relationships between these parameters. We obtained a lower percentage error with higher values of s parameter and higher number of Word2Vec features as shown in Table 2.2. We observed that the length of feature vector $v_k(t_k)$ depends on the nature of the dataset. For example in Fig. 2.6, a text document composed of a large number of words cannot be encoded completely using high number of Word2Vec features, because each visual word occupies more space in the encoded image. Moreover, we found out that error does not decrease linearly with the increase of Word2Vec features, as shown in Table 2.3.

Table 2.2: Comparison between CNNs trained with different configurations on our proposed approach. The width V (in superpixels) of visual words is fixed while the Word2Vec encoding vector size and space s (in pixel) varies. H is the height of visual word obtained.

s	V	H	w2v feat.	error (%)
4	4	1	12	7.63
8	4	1	12	5.93
12	4	1	12	**4.45**
16	4	1	12	4.83
4	4	2	24	6.94
8	4	2	24	5.60
12	4	2	24	5.15
16	4	2	24	**4.75**
4	4	3	36	6.72
8	4	3	36	5.30
12	4	3	36	**4.40**
16	4	3	36	4.77



(a) Classification error using data augmentation: (*mirror* and *crop*) over the 20news-bydate test set.

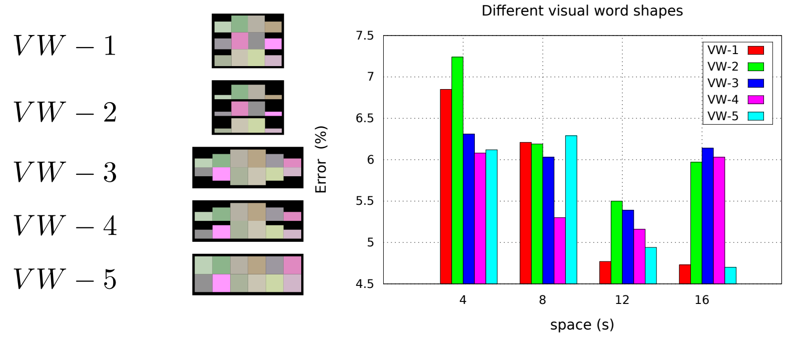


Figure 2.8: On the left, five different designs for visual words (VW) represented by 36 Word2Vec features, over the 20news-bydate dataset. The width V of these words is 4 for the first two on the top and 6 for the rest. The first four visual words consist of super pixels with different shapes to form particular visual words. On the right, a comparison over these different shapes of visual words.

We tested various shapes for visual words before selecting the best one, as shown in Fig. 2.8 (on the left). We showed that the rectangular shaped visual words obtained higher perforance as highlighted in Fig. 2.8 (on the right). Moreover, the space s between visual words plays an important role in the classification, in fact using a high value for the s parameter, the convolutional layer can effectively distinguish among visual words, also demonstrated from the results in Table 2.2. The first level of a CNN (*conv1*) specializes convolution filters in the recognition of a single superpixel as shown in Fig. 4.3. Hence, it is important to distinguish between superpixels of different visual words by increasing the parameter s.

These experiments led us to the conclusion that we have a trade-off between the number of Word2Vec features to encode each word and the number of words that can be represented in an image. In fact, increasing the number of Word2Vec features increases the space required in the encoded image to represent a single word. Moreover, this aspect affects the maximum number of words that may be encoded in an image. The choice of this parameter must be done considering the nature of the dataset, whether it is characterized by short or long text documents. For our experiments, we used a value of 36 for Word2Vec features, considering results presented in Table 2.3.

2.5.2 Data Augmentation

We encode the text document in an image to exploit the power of CNNs typically used in image classification. Usually, CNNs use "*crop*" data augmentation technique to obtain robust models in image classification. This process has been used in our experiments and we showed that increasing the number of training samples by using the *crop* parameter,

Table 2.3: Comparison of different parameters over the 20news-bydate dataset. In the leftmost table we changed the size of the encoded image from 100 × 100 to 500 × 500 and the crop size is also changed by multiplying the image size with a constant i.e. 1.13. Here *sp* stands for superpixel, *w2v* is for number of Word2Vec features, *Mw* stands for Max number of visual words that an image can contain and *#w* is the number of text documents in the test set having a greater number of words than *Mw*. We fixed the remaining non-specified parameters as follow: $s = 12$, $V = 4$, $sp = 4$, image size= 256.

image size	crop	error	sp	error
500x500	443	**8.63**	5x5	8.96
400x400	354	9.30	4x4	**8.87**
300x300	266	10.12	3x3	10.27
200x200	177	10.46	2x2	10.82
100x100	88	15.70	1x1	10.89

stride	error	w2v	Mw	#w	error
5	8.7	12	180	50%	9.32
4	8.87	24	140	64%	8.87
3	8.33	36	120	71%	**7.20**
2	**7.78**	48	100	79%	8.21
1	12.5	60	90	83%	20.66

results are improved. More precisely, during the training phase, 10 random 227 × 227 crops are extracted from a 256 × 256 image (or proportional crop for different image size, as reported in the leftmost Table 2.3) and then fed to the network. During the testing phase we extracted a 227 × 227 patch from the center of the image. It is important to note that thanks to the space *s* introduced around the encoded words, the encoding of a text document in the image is not changed by cropping. So, cropping is equivalent to producing many images with the same encoding but with a shifted position.

The "*stride*" parameter is very primary in decreasing the complexity of the network, however, this value must not be bigger than the superpixel size, because larger values can skip too many pixels, which leads to information lost during the convolution, invalidating results.

We showed that the *mirror* data augmentation technique, successfully used in image classification, is not recommended here because it changes the semantics of the encoded words and can deteriorate the classification performance. Results are presented in Fig. 2.7a.

2.5.3 Encoded Image Size

We used various image sizes for the encoding approach. Fig. 2.5 shows artificial images built on top of Word2Vec features with different sizes. As illustrated in Table 2.3, percentage error decreases by increasing the size of an encoded image; however, we observed that sizes above 300 × 300 is computationally intensive; hence, this lead us to chose an image size of 256 × 256, typically used in AlexNet and GoogleNet architectures.

Table 2.4: Testing error of our encoding approach on 8 datasets with Alexnet and GoogleNet.

Model	**AG**	**Sogou**	**DBP.**	**Yelp P.**	**Yelp F.**	**Yah. A.**	**Amz. F.**	**Amz. P.**
Xiao *et al.* [32]	8.64	4.83	1.43	5.51	38.18	28.26	40.77	5.87
Zhang *et al.* [24]	**7.64**	**2.81**	1.31	4.36	37.95	28.80	40.43	4.93
Conneau *et al.* [25]	8.67	3.18	1.29	**4.28**	**35.28**	26.57	**37.00**	4.28
Encoding scheme + AlexNet	9.19	8.02	1.36	11.55	49.00	25.00	43.75	3.12
Encoding scheme + GoogleNet	7.98	6.12	**1.07**	9.55	43.55	**24.10**	40.35	**3.01**

Table 2.5: Percentage errors on 20news-bydate dataset with three different CNNs.

CNN architecture	error
Encoding scheme + AlexNet	4.10
Encoding scheme + GoogleNet	3.81
Encoding scheme + ResNet	2.95

2.5.4 Comparison with Other State-of-the-art Text Classification Methods

We compared our approach with several state-of-the-art methods. Zhang *et al.* [24] presented a detailed analysis between traditional and deep learning methods. From their work, we selected best results and reported them in Table 2.4. In addition, we compared our results with Conneau *et al.* [25] and Xiao*et al.* [32] . We obtained state-of-the-art results on DBPedia, Yahoo Answers! and Amazon Polarity datasets, while comparative results on AGnews, Amazon Full and Yelp Full datasets. However, we obtained higher error on Sogou dataset due to the translation process [24].

It is interesting to note that the works in [24, 25] are text adapted variants of convolutional neural networks [21, 31] developed for computer vision. Therefore, we obtain similar results to these works. However, there is a clear performance gain compared to the hybrid of convolutional and recurrent networks [32].

2.5.5 Comparison with State-of-the-art CNNs

We obtained better performance using GoogleNet, as expected. We therefore believe that recent state-of-the-art network architectures, such as Residual Network would further improve the performance of our proposed approach. To work successfully with large datasets and powerful models, a high-end hardware and large training time are required, thus we conducted experiments only on 20news-bydate dataset with three network architectures: AlexNet, GoogleNet and ResNet. Results are shown in Table 2.5. We achieved better performance with powerful network architecture.

2.6 Conclusion

We presented a novel text classification approach to transform Word2Vec word embedding of text documents into encoded images to exploit CNNs models for text classification. In addition, we presented a detailed study on various parameters associated with the encoding scheme. We obtained state-of-the-art results on some datasets while in other cases our approach obtained comparative results. We showed that the CNN model generally used for image classification is successfully employed for text classification. As shown in the experiment section, the trend in results clearly show that, we can further improve results with more recent and powerful deep learning models for image classification.

3

Learning Deep Latent Space Representations

3.1 Introduction

The recent success in various computer vision tasks including visual object classification [21, 22] and speech recognition [34], demonstrated that representations play a crucial role in the performance of machine learning models. Bengio et al. [18] pointed out some of the properties of good representations including smoothness, temporal and spatial coherence, sparsity, and natural clustering. In last decade, there has been a paradigm shift in representations from hand crafted to convolutional neural networks based representations. Moreover, Razavian et. al [35] showed that CNNs based representations are more powerful than hand crafted on various image recognition tasks. Similarly, CNNs based word embeddings or representations [36] in natural language processing have replaced text representations developed on counting word occurrences. Furthermore, good multimodal representations are important for the improved performance of machine learning models. CNNs have become a de facto method to extract representation from unimodal modality [18]. and are extensively employed to represent audio, textual and visual data. In addition, these CNNs are increasingly used in the multimodal domain [37, 38, 39].

Recent years have seen a surge in tasks based on multimodal tasks including classification [4, 5, 6], semantic relatedness [8, 9], Visual Question Answering [10, 11], multimodal named entity recognition [14, 15, 16, 17], cross-modal retrieval [7, 40] and verification [41, 42]. In the existing systems, neural network based mappings have been commonly used to bridge the gap between multiple modalities in building a joint representation of each modality [43, 44]. Typically, separate networks are trained to predict features of each modality and a supervision signal is employed to reduce the distance between modalities [7, 40, 41, 42, 45, 46, 47, 48]. In addition, these works require pairwise or triplet selection at the input for training. Though by using separate networks in pairs or triplets, these systems were able to achieve good performance, however, they incur significant memory overheads. In many modern applications such as mobile devices, memory is a scarce resource therefore less memory demanding systems are required. Furthermore, both pairwise and triplet based systems suffer fromata expansion when constituting the sample pairs or sample triplets from the training set.

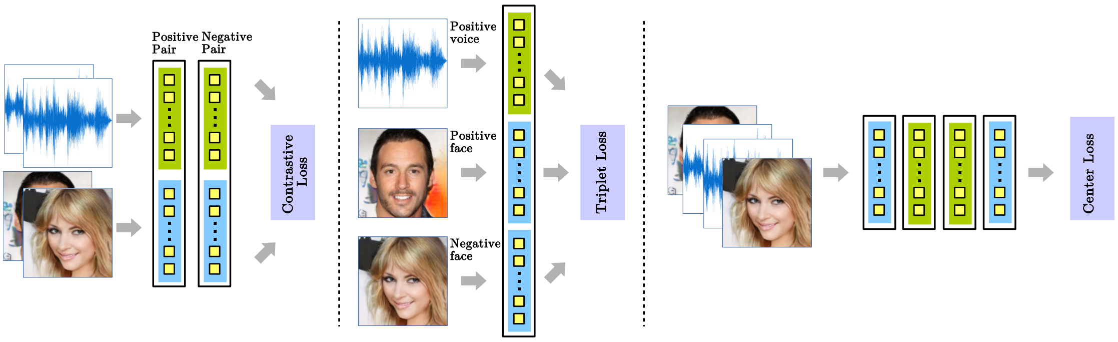
To address these issues, we introduced a deep latent space representations coupled with a single stream network (SSNet) to merge multiple modalities (text/image and audio/image) without needing a separate network for individual modality. We propose a loss function inspired from [49] as a supervision signal to map multiple modalities "nearer" to each other in the shared latent space. As a result, the proposed framework does not require complex recombination at the input. Figure 3.1 shows a generic illustration of existing systems along with the proposed framework. We employed the proposed framework in three cross-modal tasks including retrieval, matching and verification using benchmark datasets.

3.2 Related Work

Several works in the field of multimodal representation and learning have been proposed over recent years. Although each multimodal task is different from others, the underlying principle is relatively the same: to achieve semantic multimodal alignment. In this section we explore the related literature under different subsections.

One of the classical approaches towards image-text embedding is Canonical Correlation Analysis (CCA) [50]. The method finds linear projections that maximize the correlation between modalities. Works such as [51, 52] incorporate CCA to map representations of image and text to a common space. Although it is rather a classical approach, the method is efficient enough. Recently, deep CCA has also been employed to the problem of obtaining a joint embedding for multimodal data [53]. However, the major drawback is that using CCA it is computationally expensive i.e. it requires to load all data into memory to compute the covariance score.

Deep metric learning based approaches have shown promising results on various computer vision tasks. Metric learning to multimodal tasks requires within-view neighborhood preservation constraints which is explored in several works [54, 55, 56]. Triplet networks [57, 58] along with Siamese networks [59, 60] have been used to learn a similarity function between two modalities. However, most of these techniques [7, 42, 40] require separate networks for each modality which increase the computational complexity of the whole process significantly. Furthermore, these networks suffer from dramatic



(a) Learning PINs (b) Learning Image-Text Association (c) The Proposed Model

Figure 3.1: The proposed cross-modal framework (c) based on single stream network and a novel loss function to embed both modalities in the same latent space, compared with a Siamese network (a) and a Triplet network (b). In (c) the single stream network extracts the representation from both modalities while the loss function learns to bridge the gap between them.

data expansion at the input while creating sample pairs and triplets from training set.

Many different multimodal approaches employ some kind of ranking loss function as a supervision signal. Works presented in [61, 62] employ a ranking loss which penalizes when incorrect description is ranked higher than the correct one. Similarly, the ranking loss can be employed in bi-directional fashion where the penalty is based on retrieval of both modalities.

Jointly representing multiple modalities on a common space can also be employed for classification purposes. Work in [63] employs classification loss along with two neural networks for both modalities (text and image) for zero-shot learning. Work in [64] employs attention-based mechanism to estimate the probability of a phrase over different region proposals in the image. In nearly every visual question answering (VQA) method, separate networks are trained for image and text; however, [65] treats the problem as a binary classification problem by using text as input and predicting whether or not an image-question-answer triplet is correct using softmax.

In the current chapter, we extract representation from multiple modalities with a single stream network instead of two branch network without pairwise or triplet information at the input.

3.3 The Proposed Framework

The proposed framework reduces the gap between multiple modalities. The approach eliminates the need for multiple networks for each modality, since representation can be extracted with a single stream network. Figure 4.2(c) shows the proposed model.

The details of the proposed approach are presented in the following subsections. In Subsection 3.3.1, 3.3.2 and 3.3.3, we explain various mechanisms employed to extract the representations of text, audio and image modalities. While Subsection 3.3.4 provides details of the single stream network along with the objective function to bridge the gap between various modalities.

3.3.1 Encoding Text Descriptions

Semantics plays a crucial role to understand the meaning of a text description. Humans can understand semantics easily, however the same performed automatically becomes a challenging task. Word2Vec word embedding [36] takes one step towards mathematically representing the semantic relationships between words. Its objective function causes words that occur in a similar context to have similar word embedding. We propose presented an encoding scheme exploiting Word2Vec to reconstruct the semantics associated with a text description as an image [66]. We employ this encoding scheme to encode text descriptions as images and use these encoded text images as input to the neural networks originally developed for image input. We explain the encoding scheme to transform a text description into an image in Figure 3.2. The encoding scheme extracts the Word2Vec emdedding of a word, then normalize each component to assume values in the interval $[0 \ldots 255]$. Finally, the normalized values are interpreted in triplets as RGB sequence. This encoding scheme enabled us to use a single stream network for both text to image and image to text retrieval. The single stream network based cross-modal retrieval has potential for memory efficient and computationally-inexpensive applications on low powered devices.

3.3.2 Audio Signals

In addition to the encoded text input, audio signals are also fed to the network. The encoded audio signals are short term magnitude spectrograms generated directly from raw audio of length three seconds. The audio stream is extracted, converted to a single channel at 16 kHz sampling rate, spectrograms are then generated in a sliding window fashion using a hamming window [67, 41]. The generated spectrograms are used as input to a standard neural network.

3.3.3 Visual Signals

The input to the proposed single stream network consists of three channel (RGB) image. Other modalities (audio and text) are transformed into similar visual signals through encoding schemes mentioned earlier in Subsection 3.3.1 and 3.3.2. These encoding schemes are extremely helpful to employ the the single stream network.

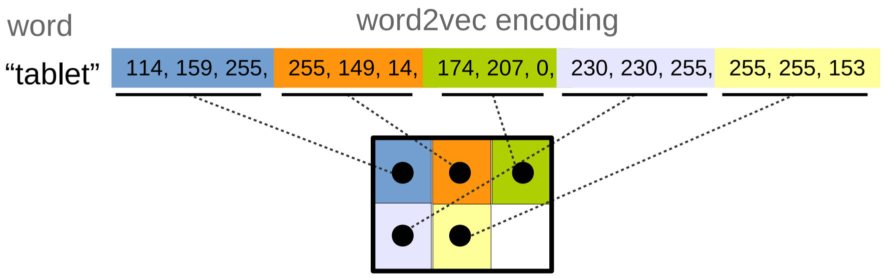


Figure 3.2: The word "tablet" is encoded into an image using Word2Vec encoding with vector length 15. Consecutive words in the text descriptions are encoded as image preserving relative position of each word. Note that words that occur in similar context will have similar embedding, thus the encoding will be similar in color space. (Best viewed in color)

3.3.4 The Single Stream Network

The deep latent space representation and learning framework coupled with a single stream network extracts features from multiple modalities to minimize intra and cross modality variations. Suppose there are n_s samples of a modality associated with n_i samples of second modality in a class c. Data from both modalities is input to the network and $n_s + n_i$ feature vectors f_c are obtained at the output of the network. During training, geometric centers of $n_s + n_i$ feature vectors is computed and the objective function consisting of the distance d of each feature vector from the center, is minimized for all the classes.

$$d(f_c) = \sum_{i=1}^{n_s+n_i} \| f^i - \frac{1}{n_s + n_i} \sum_{j=1}^{n_s+n_i} f^j \|_2^2 \tag{3.1}$$

Thus, during the training phase, data from both modalities is treated in similar fashion and the proposed single stream network can effectively bridge the gap between two modalities removing the need of multiple networks. In the implementation, instead of using the traditional loss functions, we extend center loss for learning deep latent space jointly trained with softmax loss [49]. This loss function simultaneously learns centers for all classes and minimizes the distances between each center and the associated samples from both modalities. It thus imposes neighborhood preserving constraint within each modality as well as across modalities. If there are n classes in a mini batch M with m samples, the loss function is given by

$$\mathcal{L}(\mathrm{M}) = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T f^i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T f^i + b_j}} + \frac{\lambda}{2} \sum_{c=1}^{n} d(f_c) \tag{3.2}$$

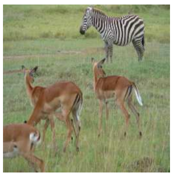
Image	Text Description
	– A few deer and a zebra on a grass field. – A few gazelles near a zebra in a field. – A bunch of animals that are in some grass. – The zebra is standing near four brown deer. – Some gazelle and a zebra standing in a field.

Figure 3.3: A random example taken from MSCOCO dataset. The dataset consists of samples made of an image and 5 representative text descriptions.

In Eq. 3.2, $f^i \in \mathbb{R}^d$ denotes the i-th deep feature, belonging to the y_j-th class. d is the feature dimension. $W_j \in \mathbb{R}^d$ denotes the j-th column of the weights $W \in \mathbb{R}^{d \times n}$ is the last fully connected layer and $\mathbf{b} \in \mathbb{R}^n$ is the bias term. A scalar λ is used for balancing the two loss functions. The conventional softmax loss can be considered as a special case of this joint supervision, if λ is set to 0 [49].

This loss function minimizes the variation between the two modalities and effectively preserves the neighborhood structure. In this way, modalities which do not belong to the same class do not occur in the same neighborhood. The proposed framework is generic and an appropriate deep network can be employed to extract representations from both modalities. In the implementation, InceptionResNet-V1 is used as a single stream network for joint embedding of two modalities (text/image or audio/image).

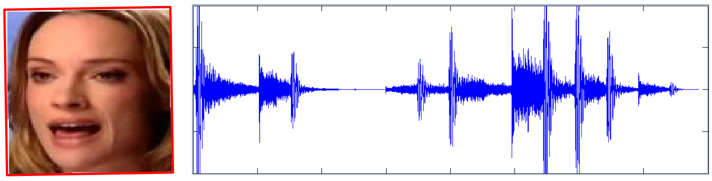
3.4 Experiments

We perform series of experiments on various tasks consisting of *cross-modal retrieval*, *matching* and *verification* to evaluate the embedding learned by the single stream network under the proposed framework. The experimental setup and dataset details are explained below.

3.4.1 Datasets

We evaluate the proposed framework on two publicly available benchmark datasets including MSCOCO [68] and VoxCeleb [67]. MSCOCO dataset is employed for cross-modal task on text/image, while VoxCeleb dataset is used for cross-modal matching and verification tasks on audio/image. MSCOCO dataset contains $123,287$ images, and each image is annotated with 5 captions. Figure 3.3 shows a random example selected from MSCOCO dataset. We use 1000 images for testing and the rest for training as proposed by the original authors [68] and used by Wang *et al.* [7] and referred it as COCO-1k.

VoxCeleb is an audio-visual dataset consisting of short clips of human speech, ex-



Face Audio

Figure 3.4: An audio-visual example extracted from Voxceleb dataset.

tracted from interview videos uploaded to YouTube [67]. Figure 3.4 shows an example of audio-visual information taken from VoxCeleb dataset. We two train/test splits out of this dataset to perform various cross modal tasks as recommended by [42]. The first split consists of disjoint videos from the same set of speakers while the second split contains disjoint identities. We train the model using two training sets, allowing us to evaluate on both test sets, the first one for *seen-heard* identities and the second for *unseen-unheard* identities.

3.4.2 Experimental Setup

We perform three different experiments which are as below.

3.4.2.1 Cross-modal Retrieval

In the first task, we evaluate the learned embedding on *retrieval* with MSCOCO dataset. Given a single modality input, the task is to retrieve all the semantic matches of the opposite modality. We perform this task for both Image → Text and Text → Image formulation. For the sake of comparison with other techniques, we use $R@K$ metric as described in [69]. We employ the $R@1$, $R@5$ and $R@10$ which means that the percentage of queries in which the first 1, 5 and 10 items are found in the ground truth.

3.4.2.2 Cross-modal Verification

The second task is to perform *verification* on VoxCeleb dataset where the goal is to verify if audio segment or a face image belong to the same identity. Two inputs are considered i.e. face and voice and verification between the two depends on a threshold on the similarity value. The threshold can be adjusted in accordance to wrong rejections of true match and/or wrong acceptance of false match. We report results on verification metrics i.e. ROC curve (AUC) and Equal Error Rate (EER).

3.4.2.3 Cross-modal Matching

Finally, the last task consists of *matching* on VoxCeleb dataset where the goal is to match the input modality (probe) to the varying gallery size n_c which consists of the other modality. We increase n_c to determine how the results change. For example, the 1 : 2 task, we are given a modality at input, e.g. face, and the gallery consists of two inputs from other modality, e.g. audio. One of them contains a true match and other serves as an imposter input. We employ matching metric i.e. accuracy to report results. We perform this task in five settings where in each setting the n_c is increased i.e. $2, 4, 6, 8, 10$.

3.4.3 Implementation Details

We learn the proposed single stream network with standard hyper-parameters setting. The size of the input image and encoded text is set to 128×128 on MSCOCO dataset for retrieval task while the input image and spectrogram is set to 256×256 on VoxCeleb dataset for verification and matching tasks. The output feature vector is $128-d$ extracted from the last fully connected layer of the single stream network. For optimization we employ Adam optimizer [70] because of its ability to adjust the learning rate during training. We use Adam's initial learning rate of 0.05 and employ weight decay strategy with decaying by a factor of 5e−5. Two networks are trained for 100 epochs on MSCOCO and VoxCeleb. The mini-batch size was fixed to randomly select 45 samples from the training set.

3.5 Evaluation

3.5.1 Cross-modal Retrieval

In this section we evaluate the results of *cross-modal retrieval* task employing both text and image as probe with other modality at the retrieval end. We report results in terms of $R@K$ metric which evaluates the top K retrieved results. Table 3.1 demonstrates quantitative results of our approach on the said task. Compared to the current state-of-the-art, our proposed framework performance is comparatively low. The main reason is due to the fact that $R@K$ is based on whether query's pair appeared or not in the retrieval result. So, even if retrieval result is semantically similar and if query's pair did not appear in the retrieval result, the $R@K$ score is considerably low.

3.5.2 Cross-modal Verification

In this section we report results of the framework on *cross-modal verification* task, the aim of which is to determine whether an audio segment and a face image are from

Table 3.1: Comparison of the proposed framework with current state-of-the-art methods on cross-modal retrieval using $R@K$ measure on MSCOCO dataset.

	MSCOCO					
Model	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
DVSA [12]	38.4	69.9	80.5	27.4	60.2	74.8
HM-LSTM [71]	43.9	-	87.8	36.1	-	86.7
m-RNN-vgg [72])	41.0	73.0	83.5	29.0	42.2	77.0
Order-embedding [69]	46.7	-	88.9	37.9	-	85.9
m-CNN(ensemble) [73]	42.8	73.1	84.1	32.6	68.6	82.8
TextCNN [74]	13.6	39.6	54.6	10.3	35.5	55.5
FV-HGLMM [74]	14.3	40.5	55.8	12.7	39.0	57.2
Str. Pres. [7]	**50.1**	**79.7**	**89.2**	**39.6**	**75.2**	**86.9**
Proposed SSNet	40.0	64.4	76.7	30.9	62.7	73.7

the same identity or not. Recently [42] used VoxCeleb dataset to benchmark this task under two evaluation protocols, one for *seen-heard* identities and the other for *unseen-unheard* identities. We evaluate on the same test pairs[1] created in [42] for each evaluation formulation. More specifically, $30,496$ pairs from *unseen-unheard* identities and $18,020$ pairs from *seen-heard* identities are selected. The results for cross-modal verification are reported in Table 3.2. We use AUC and EER metrics for verification. As can be seen from the table, our model trained from scratch outperformed the state-of-the-art work on *seen-heard* and *unseen-unheard* protocols.

Furthermore, we examine the effect of Gender (G), Nationality (N) and Age (A) separately, which influence both face and voice verification. It is important to note that [42] employed pre-trained network, whereas we trained the model from scratch. Our network outperformed on G, N, A and the combination (GNA) in *seen-heard* formulation regardless of pre-trained network as a backbone, see Table 3.3. However, our network shows comparable results on *unseen-unheard* formulation for N,A and GNA, whereas it outperformed on random and G regardless of pre-trained network, see in Table 3.3.

3.5.3 Cross-modal Matching

In this section we perform the *cross-modal matching* task employing the framework. We perform the $1 : n_c$ where $n_c = 2, 4, 6, 10$ matching tasks to evaluate the performance of our approach. Unlike others [41, 42], we do not require positive or negative pair selection since under the proposed framework, the network learns in a self-supervised manner.

[1] `http://www.robots.ox.ac.uk/vgg/research/LearnablePins`

Table 3.2: Cross-modal verification results on *seen-heard* and *unseen-unheard* configurations with model trained from Scratch.

	AUC %	EER %
	Seen-Heard	
Learnable Pins [42]	73.8	34.1
Proposed SSNet	**91.1**	**17.2**
	Unseen-Unheard	
Learnable Pins [42]	63.5	39.2
Proposed SSNet	**78.8**	**29.5**

Table 3.3: Analysis of cross-modal biometrics under varying demographics for seen-heard and unseen-unheard identities. Note that SSNet has produced best results when trained from scratch.

Demographic Criteria	Configuration	Random	G	N	A	GNA
		Seen-Heard (AUC %)				
Learnable Pins [42]	Scratch	73.8	-	-	-	-
Learnable Pins [42]	Pre-train	87.0	74.2	85.9	86.6	74.0
Proposed SSNet	Scratch	**91.2**	**82.5**	**89.9**	**90.7**	**81.8**
		Unseen-Unheard (AUC %)				
Learnable Pins [42]	Scratch	63.5	-	-	-	-
Learnable Pins [42]	Pre-train	78.5	61.1	77.2	**74.9**	**58.8**
Proposed SSNet	Scratch	**78.8**	**62.4**	53.1	73.5	51.4

Table 3.4 reports results on said task along with comparison with recent approaches on the same task. In Table 3.4, the probe is voice while the matching gallery consists of faces. For instance consider the case where the input is voice and is 1 : 2 matching task, we figure out the entry in gallery that matches the input. It is important to note that for $n_c > 2$ tasks, the work in [41] trains separate network for each n_c. However, the major advantage of training under the proposed framework is that it is not restricted by increasing n_c. Thus, the proposed single stream network can effectively handle n_c size without increasing sub-network size to categorize each n_c. However, increasing n_c decreases performance in a linear fashion due to increase in challenge.

Table 3.4: Accuracy score of cross-modal forced matching task comparing Learnable PINS [42], SVHF-Net [41] and SSNet.

Inputs	Learnable PINS.	SVHF-Net	Proposed SSNet
Voice → Face (%)			
2	84	78	78
4	54	46	56
6	42	39	42
8	36	34	36
10	30	28	30

3.5.4 Qualitative Evaluation

Figure 3.5 is t-SNE [75] visualization of features from MSCOCO test set i.e. 1k images with five text descriptions for each image. Once the network is trained on the dataset, features of the test set are extracted from the model and are fed to t-SNE, Visualization verify that the proposed framework is capable of bridging gap between image and encoded text description in the latent space. The image and text encoding description are overlapped and distributed enough for being discriminated in retrieval. Some bidirectional retrieval results from MSCOCO test set are also present in Figure 3.6. It can be seen that query and retrieved objects are related.

Figure 3.6 is tSNE [75] embedding result of learned features extracted from test set of VoxCeleb dataset for 10 identities. We visualize learned embedding for both formulations i.e. *seen-heard* and *unseen-unheard*. Visual illustrations support the hypothesis that no pair selection knowledge at pre/post processing stage is required for network to learn mapping of identities in shared latent space. Given the faces and voices at input, the network learns to map both modalities conditioned at class information due to formulated loss function in Eq. 3.2. Note that, unseen-unheard formulation has very high level of difficulty even though results shown in Figure 3.6 are impressive.

3.5.5 Ablation Study

We experiment with the hyperparameter λ which is used to couple conventional softmax with the proposed loss signal, see Eq 3.2. When the value of λ is fixed to 0, the loss function is a special case where only softmax's penalization is employed. Increasing value of λ introduce the increasing effect of coupled penalization. We experiment with two values of λ where we set it to 0 and 1 for two evaluation protocol of cross-modal verification and forced matching. The quantitative scores are reported in Table 3.5 and 3.6. In these experiments, only λ is varied otherwise joint formulation configuration is same as for previous experiments. These experiments explain the crucial need for

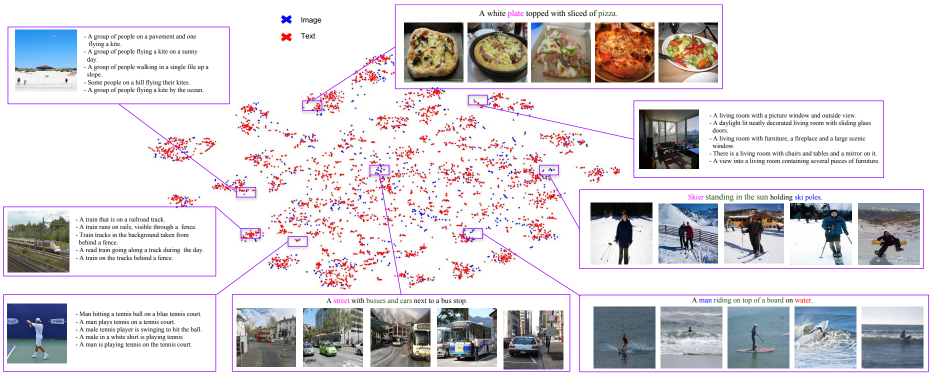
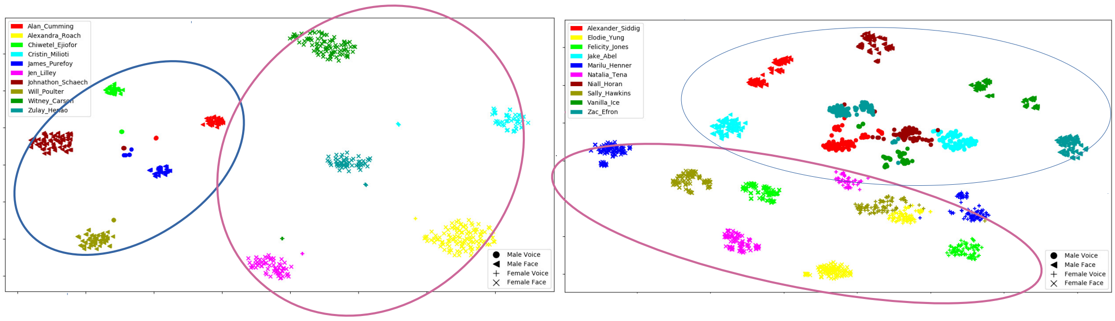


Figure 3.5: Embedding of MSCOCO test set in latent space visualized using t-SNE [75]. The similar text feature vectors (shown in red) and the image feature vectors (shown in blue) are close in the embedding space. Few bidirectional retrieval results are also shown which are similar. (Best viewed with color)



Seen-Heard

Unseen-Unheard

Figure 3.6: Visualization of learned voice and face embedding extracted from test set of the VoxCeleb dataset for 10 identities. The pink oval encloses female entities while the male entities are enclosed in blue one. (Best viewed in color)

penalization beyond softmax in the proposed setting and establish the effectiveness of penalization based on centers for tasks such as verification, matching and retrieval.

3.6 Conclusion

In this chapter, a framework is proposed for deep latent space representation and learning based on a single stream network for multimodal applications. The proposed framework is applied on three cross-modal tasks including retrieval, verification and matching on

Table 3.5: Cross-modal Verification results on *seen-heard* and *unseen-unheard* configurations to illustrate the effect of proposed loss function.

Configuration	AUC %	EER %
	Seen-Heard	
$\lambda = 0.0$	81.2	26.3
$\lambda = 1.0$	**91.1**	**17.2**
	Unseen-Unheard	
$\lambda = 0.0$	72.6	33.6
$\lambda = 1.0$	**78.8**	**29.5**

Table 3.6: Accuracy score of cross-modal forced matching task to illustrate the effect of proposed loss function.

Inputs	$\lambda = 0.0$	$\lambda = 1.0$
	Voice → Face (%)	
2	73	78
4	49	56
6	38	42
8	34	36
10	29	30

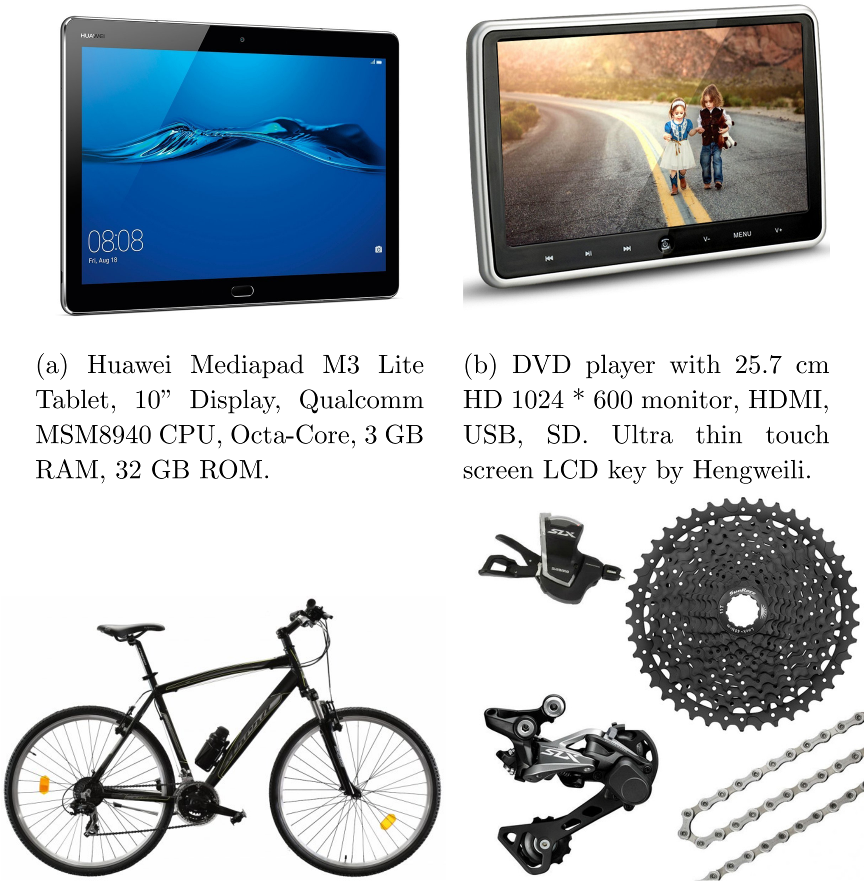
text/image and audio/image modalities. The proposed framework was able to reduce the gap between different modalities by learning a shared latent space. Thus the framework can generate discriminative representations of various modalities. The proposed framework requires encoding schemes to transform text or audio signals to images. The performance of the proposed system may increase if better encoding schemes are learnt. In future, we will investigate other encoding schemes more suitable for single stream networks. One of the core strengths of reliance on learning features in a shared latent space is no overhead of pair or triplet selection at the input. As dataset increases exponentially over time, so does the overhead of pairs or triplets selection in the existing methods. The proposed framework ensures that class information is leveraged to penalize distance between learned embedding. We achieved state-of-the-art results for verification and matching on VoxCeleb dataset while promising results on MSCOCO dataset.

4 Learning Deep Fused Representation

4.1 Introduction

Information in real-world applications usually comes from multiple sources. Images are often associated with tags or captions; for example in the world of e-commerce products on sale are displayed using one or more images with one or more text descriptions such as product title, summary and technical details. Each source is characterized by distinct statistical properties that makes it difficult to create a joint representation that uniquely captures the "concept" in the real-world. For example, Figure 4.1 shows four advertisements, where, in the first row, two objects have seemingly similar images but different text descriptions, conversely, in the second row, we have two different images but similar text descriptions. For an image classification model it would not be easy to distinguish two images on the first row while it would have no difficulty in distinguishing images on the second one. Similarly, for a text classification model it would be difficult to classify two text descriptions on the second row while it would have no difficulty in classifying correctly descriptions shown on the first one. Such scenarios present a challenge to create a joint representation of an image and associated text description. This leads us to create a representation for such classification problem. This representation can exploit such scenarios to remove ambiguity and improve classification performance.

The use of joint representation based on image and text features is extensively employed on a variety of tasks including modeling semantic relatedness, compositionality, classification and retrieval [76, 77, 78, 9, 5]. Typically, in such approach, image features are extracted using CNNs. Whereas, to generate text features, Bag-of-Words models or Log-linear Skip-gram Models [79] are commonly employed. This represents a challenge to find relationships between features of multiple modalities along with representation, translation, alignment, and co-learning as stated in [80]



(a) Huawei Mediapad M3 Lite Tablet, 10" Display, Qualcomm MSM8940 CPU, Octa-Core, 3 GB RAM, 32 GB ROM.

(b) DVD player with 25.7 cm HD 1024 * 600 monitor, HDMI, USB, SD. Ultra thin touch screen LCD key by Hengweili.

(c) Men's hybrid bicycle with aluminum frame and **Shimano SLX M7000 11-speed gearbox**.

(d) **Shimano SLX M7000 11-speed gearbox** with derailleur gears and chain.

Figure 4.1: In the upper row, two examples of ambiguous images that can be disambiguated through analysis of the respective text description. In the lower row, two examples of ambiguous text description that can be disambiguated through analysis of respective images.

Traditionally, there are two general strategies for text and image fusion referred to as early and late fusion [81, 80]. In early fusion [82, 78], features from each modality are concatenated in a single vector and fed as input to a classification unit. In contrast,

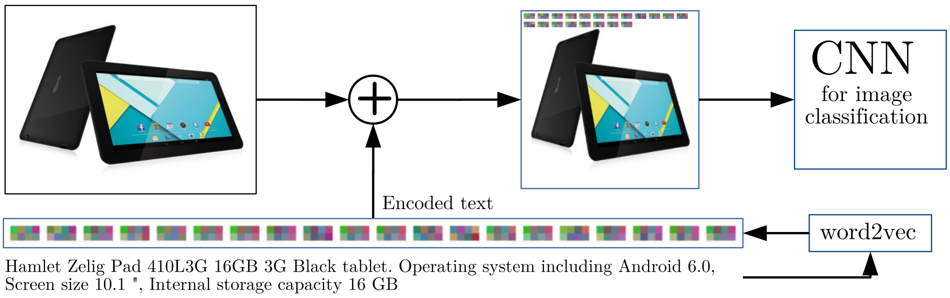

CNN
for image
classification
Encoded text
word2vec
Hamlet Zelig Pad 410L3G 16GB 3G Black tablet. Operating system including Android 6.0,
Screen size 10.1 ", Internal storage capacity 16 GB

Figure 4.2: The proposed encoded text and image fusion model for deep multimodal classification. The text is encoded within the image so that the CNN model can exploit semantics along with the information of the image.

late fusion [6, 83] uses decision values from each classification unit and fuses them using a fusion mechanism employing a weight sharing strategy. The work in [6] showcases a comparative study of multimodal fusion methods to perform multimodal classification in real-world scenarios. Specifically, in [6], late fusion produced better performance compared to early fusion method, however, late fusion comes with the price of an increased learning effort. Recently, [5], fuses data from discrete (text) and continuous (image) domains and showcases the efficiency of fusion strategies in terms of learning and computational expense. Our approach is similar to early fusion strategy, where a single classifier is needed to perform multimodal classification, as stated in [81, 80]. However, we concatenate encoded text features into an image to obtain an information enriched image. Finally, an image classification model is trained and tested on these images. Intuitively, concatenating text descriptions onto images may not sound motivating due to several reasons. Since the idea is overlaying the encoded tex tdescription into an image, it might affect the image perception in general. However, we observed that the joint representation of encoded text and image improves the multimodal classification.

With this work, we present a novel strategy which combines a text encoding schema to fuse image and text in a information enriched image. The encoding schema is based on Word2Vec word embedding [36] that transforms embedding to encoded text. We fuse both text description and image into a single source so that it can be used with a CNN architecture. We demonstrate that by adding encoded text information in images better classification results are obtained compared to the best one obtained using a single modality.

4.2 The Proposed Approach

Multimodal strategies fuse image and text description into integrated representation. We obtain transform Word2Vec word embedding [36] into visual embedding from previous

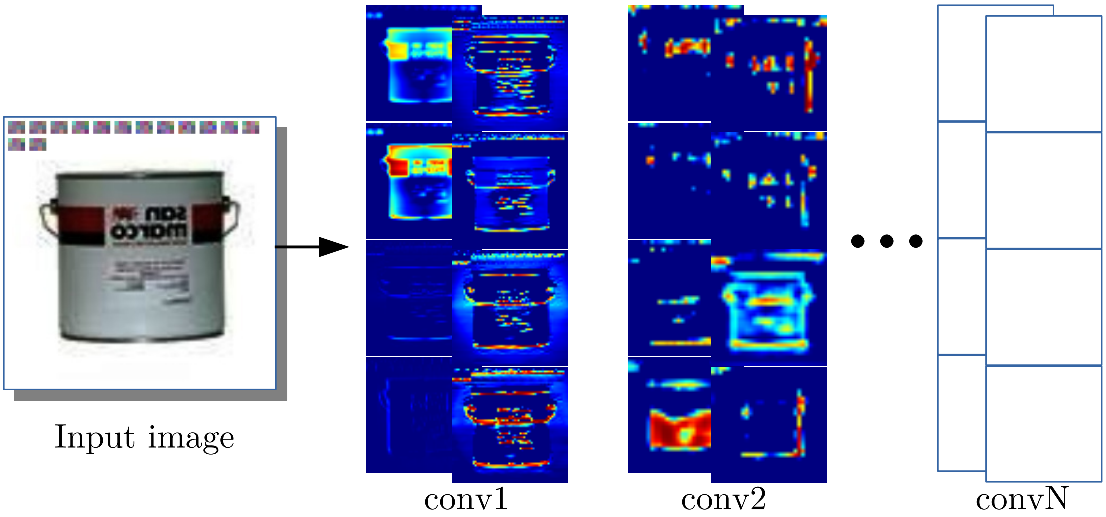


Figure 4.3: An example of an input image and some feature maps of the first two convolutive layers of the CNN used for the Ferramenta dataset. These examples of feature maps show significant activations on both textual information and image details.

Chapter 2. However, we fuse encoded text with associated image to obtain an information enrich image. An example of information enriched image is shown in Figure 4.2. This image can be fed to a CNN configured for image classification to learn multimodal representations. In other words, multimodal classification problem is transformed into a typical image classification task. Finally, state-of-the-art image classification network can be employ with softmax function. Figure 4.3 shows the behavior of a CNN that receives a joint representation with our approach. In the same figure we can notice how some convolutive filters of the first two layers, are activated both on the image and on the encoded text description. This approach is suitable to be adopted in a multimodal strategy because a CNN model can extract information from both sources (Text/Image).

4.3 Experiments and Results

4.3.1 Datasets

Typically, multimodal dataset consists of an image and associated text description. In this work, we use three large scale multimodal datasets to show the applicability of our approach to various domains.

The first dataset is named Ferramenta multimodal dataset [66]. This dataset is made up of 88, 010 adverts divided in 66, 141 adverts for train and 21, 869 adverts for test, belonging to 52 classes. Ferramenta dataset provides a text and representative image for each commercial advertisement. It is interesting to note that text descriptions in this dataset are in Italian Language.

Another dataset used is the UPMC Food-101 [84], a large multimodal dataset containing about 100, 000 items of food recipes classified in 101 categories. This dataset was crawled from the web and each item consists of an image and the HTML webpage on which it was found. We have only extracted the title from every HTML document.Categories in the dataset are the 101 most popular categories from the food picture sharing website[1].

We used another publicly available real-world multimodal dataset called Amazon Product Data [85]. The dataset consists of advertisements with each advertisement contain a text description and image. We randomly selected 10, 000 advertisements belonging to 22 classes. Finally, we split 10, 000 advertisements for each class into train and test sets with 7, 500 and 2, 500 advertisements respectively.

We applied *mirroring* and *cropping* to these datasets. To avoid losing the semantics of the encoded text, we applied above mentioned techniques directly on images, before merging them with the encoded text descriptions.

4.3.2 Implementation Details

In this work, the transformed text description is fused into original, horizontally flipped and cropped version of an image with 256 × 256 pixel size. Some examples are shown in Figure 4.4. We use a standard AlexNet [21] and GoogleNet [86] with softmax as supervision signal. We use the following hyperparameters for both networks: learning rate $lr = 0.01$, solver type = Stochastic gradient descent (SGD), training epochs = 90 and/or till no further improvement is noticed to avoid over fitting. In our experiments, accuracy is used to measure classification performance. We conducted five fold experiments on each dataset to evaluate the proposed approach.

4.3.3 Experiment Details

In the first set of experiments we extract only images from the three datasets with an image size of 256 × 256, then we train a standard AlexNet and GoogleNet from scratch. Results are shown in the column labelled “Image” of Table 4.1. In the second set of experiments, we use text descriptions and train a Word2Vec model to extract feature vectors, which then have been used as input to train a Support Vector Machine (SVM). Results are shown in the column named “Text” of Table 4.1. Later, we use images and text descriptions to create information enriched images with the proposed approach. Results are shown in the last column of Table 4.1. Analyzing results in Table 4.1 it can be noticed that the proposed method outperforms the accuracy of the best results obtained using only text descriptions or images on all three datasets. Images in the

[1] www.foodspotting.com

Table 4.1: Classification results and comparisons with the CNN trained on single source (column Image) along with images and text descriptions fused using the proposed approach (column Proposed). Note that column Text shows results of a SVM trained on Word2Vec features.

Dataset	Image	Text	**Proposed**
		AlexNet	
Ferramenta	92.36	84.50	**94.84**
Amazon Product	46.07	64.37	**72.52**
Food-101	42.01	56.75	**83.04**
		GoogleNet	
Ferramenta	92.47	84.50	**95.87**
Amazon Product	51.42	64.37	**78.26**
Food-101	55.65	56.75	**85.69**

Ferramenta dataset contain objects on a white background, this explains the excellent classification result obtained on images alone. On the contrary, images in the UPMC Food-10 dataset are with complex background and extracted from different contexts, which leads to a low classification performance of the images without text. Results of our approach when applied to the UPMC Food-10 and Amazon Product Data datasets, highlight the strengths of our approach: the fusion of two very different information into a single image space exploits the two types of information content in the best way. From the Table 4.1, it is evident that the proposed approach obtains higher classification performance with GoogleNet compared with AlexNet. With this result, we expect higher multimodal classification performance using the recent state-of-the-art CNNs for image classification.

4.3.4 Baselines

In a multimodal setting, text and image are two standard baselines [6, 5, 37] to which different fusion strategies are compared. In our work, we use Word2Vec word embedding as text baseline and standard image classification model as image baseline. Finally, we compare our fusion approach with these baseline strategies, results are shown in Table 4.2. Our approach obtains higher classification accuracy compared to the unimodal (Text/Image).

Table 4.2: Comparison of our approach with baseline and previous available works.

	Model	Ferramenta	UPMC Food-101	Amazon Product Data
Previous work	Wang et. al [84]	–	85.10	–
	Kiela et. al [5]	–	**90.8±0.1**	–
	Gallo et. al [6]	94.42	60.63	–
Baseline	Image	92.47	55.65	51.42
	Text	84.50	56.75	64.37
Ours	Proposed	**95.87**	85.69	**78.26**

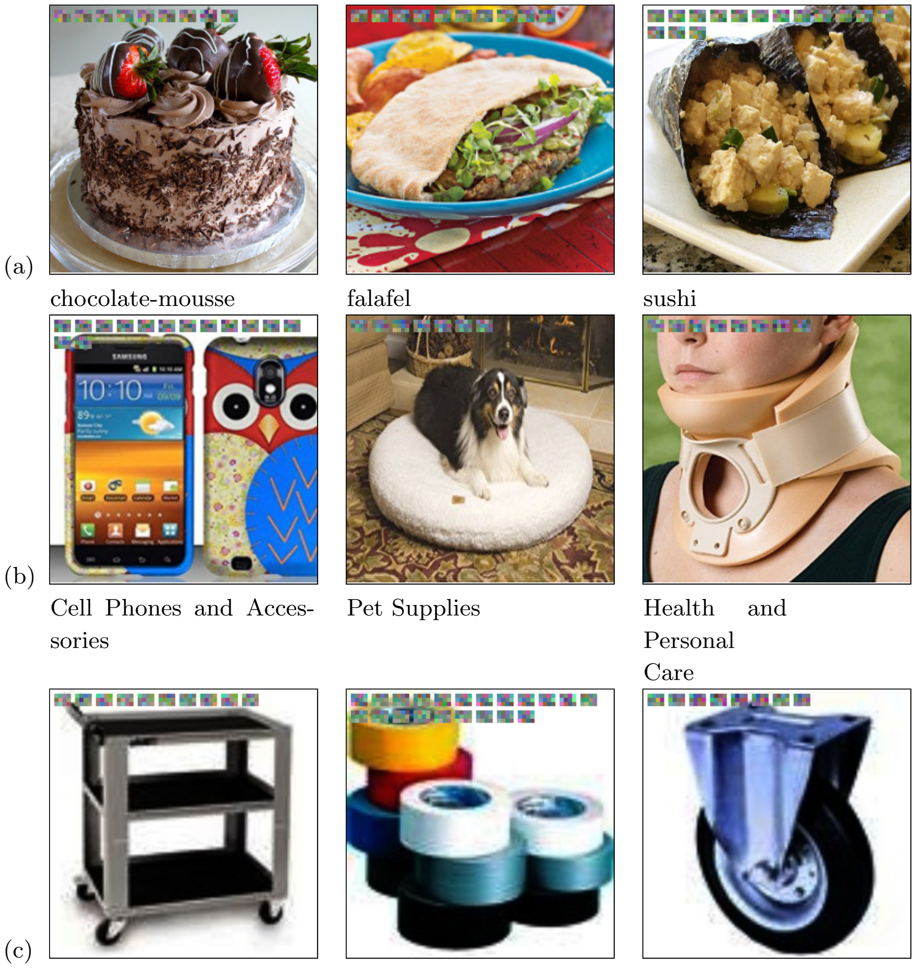
4.3.5 Comparison with State-of-the-Art Methods

In addition, we compare our fusion approach with state-of-the-art multimodal works available in literature, results are shown in Table 4.2. Our approach obtains higher or comparative classification accuracy compared to previous available works. We also include results from [84] on UPMC Food-101, where they used TF-IDF features for text and a deep convolutional neural network features for images. Furthermore, we compare our work with [5] where they explore the trade off between efficiency and multiple fusion strategies. Additionally, we compare our results with [6] where they use Word2Vec and Bag-of-Words for text and a deep convolutional neural network for images. We note that in the case of Ferramenta, our method works considerably better than previously reported results. We obtain comparable results on UPMC Food-101 dataset. We find that [6] is not scalable, whereas our work can be employed regardless of the dataset size avoiding any bottlenecks.

4.4 Conclusion

In this chapter, we proposed a new approach to fuse images with their text description so that a CNN architecture can be employed as a multimodal classification unit. To the best of our knowledge, the proposed approach is the only one that simultaneously exploits text semantics and image casted to a single source, making it possible to use a single classifier typically used in standard image classification. The classification accuracy achieved using our approach maintains an upper bound to single modalities.

Another very important contribution of this work concerns the joint representation into the same source of two heterogeneous modalities. This aspect paves the way to a still open set of problems related to the translation from one modality to another where relationships between modalities are subjective.



(a) chocolate-mousse falafel sushi

(b) Cell Phones and Accessories Pet Supplies Health and Personal Care

(c) Trolley Belt Wheel

Figure 4.4: Some random examples of multimodal fusion from the datasets used. In row (a) three examples of images with the corresponding classes, extracted randomly from the UPMC Food-101 dataset, while in row (b) the three images were extracted from the Amazon Product Data dataset and in (c) three example images extracted from the Ferramenta Dataset. The size of all images is 256 × 256. The text associated to each image is encoded and visually represented in the upper part of the image.

5
Inward Scale Feature Representations

5.1 Introduction

In last few years, mainly due to advances in CNNs the performance on tasks such as image classification [22], cross-modal and uni-modal retrieval [7, 74] and face recognition and verification [87, 49, 88] has increased drastically. It has been observed that deeper architectures tend to provide better capabilities in terms of approximating any learnable function. A common observation is that these architectures (with large number of parameters) can learn features at various levels of abstraction. However, it is a well-know that these architectures are more prone to overfitting than shallower counterparts, thus hampering generalization ability, furthermore deep architectures are computationally expensive. Majority of CNNs based pipelines follow the same structure i.e. alternating convolution and max pool layers, fully connected along with activation functions and dropout for regularization [89, 23, 30].

Discriminative learning techniques [90, 60, 49] aiming to embed the learned feature representations onto a hyperspace, linear or quadratic spaces has remarkably increased the performance of deep learning architectures. There are studies in literature [91, 92] arguing that in higher dimensions when the data is projected onto an input space divergence is reduced in terms of distance ratio between the nearest and farthest neighbors to a given target and tends to be ≈ 1. Due to this reduced contrast distance, input point cannot be discriminated effectively. It is important to note that since retrieval and search tasks tend to operate on higher dimensions, this phenomenon is valid for these tasks as well. Euclidean distance can be formulated as $L_2 = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ where

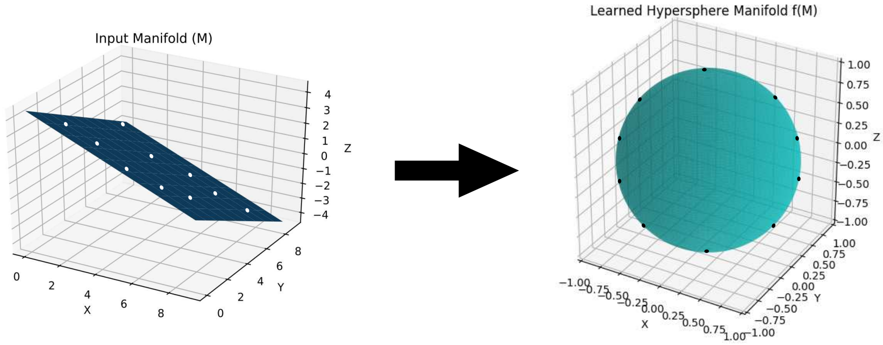


Figure 5.1: A toy example representing how the projection takes place. The manifold M is transformed into a hypersphere $f(M)$ during training. The small black and white dots represent different classes. The perfect alignment of all the classes on the circumference of the hypersphere is ideal condition assuming that there exists no intra-class variation. (best viewed in color)

x_i, x_j and y_i, y_j are two points in the input space. Surprisingly enough, [91] argues that in L_k-norms, the meaningfulness in high dimensionality is not independent of value of k with lower values of k norms performing better than their greater value counterparts i.e. $L_2 < L_1$. The general formula of L_k norm can be setup as $L_k(x, y) = \sum_{i=1}^{d}(\|x_i - y_i\|^k)^{1/k}$ for $k = 1, 2, 3, ..., n$. The relation considers norms with $k = \frac{1}{2}, \frac{1}{3} ... \frac{1}{n}; \forall k < 1, n \in Z$, referred to as fractional norms. Although fractional norms do not necessarily follow the triangle inequality $L_k(x, z) \leq L_k(x, y) + L_k(y, z); \forall x, y, z \in X$ where X is the input space, they tend to provide better contrast than their integral counterparts in terms of relative distances between query points and target.

In the current chapter, we explore projections of feature representations onto different hyper-spaces and propose that hypersphere projection has superior performance to linear hyperspace where discriminative analysis and disintegration of multiple classes becomes challenging for deep architectures, as shown in Figure 5.1. We propose that inward scaling applied to projections on a hypersphere enhances the network performance in terms of classification and retrieval. We evaluate the proposed inward scaling layer on a number of benchmark datasets for classification and retrieval. We employ MNIST, FashionMNIST [93], CIFAR100 [94] and SVHN [95] datasets for classification while FashionMNIST is used for retrieval. Note that the inward scaling layer is not dependent on a particular deep architecture and it can be applied to different networks including VGG, Inception-ResNet-V1, GoogleNet [22] and can be trained in an end-to-end fashion.

5.2 Related Work

5.2.1 Metric Learning

Metric learning aims at learning a similarity function which can also be referred as a distance metric. Traditionally, metric learning approaches [96, 97, 98] focused on learning a similarity matrix M_i between two vectors. Consider feature vectors $X = (x_1, x_2, ..., x_n)$ where each vector x_i corresponds to the relevant features. Then the similarity matrix for a corresponding distance metric can be computed as $\|x_i - x_j\| = \sqrt{(x_i - x_j)^T M_i (x_i - x_j)}$ where x_i and x_j are given features. However, in recent metric learning methodologies [54, 99, 100, 101, 102], neural networks are employed to learn the discriminative features followed by a distance metric $d(x_i, x_j)$. Contrastive loss [103, 101] and Triplet loss [57, 58, 60] are commonly used metric learning techniques. Contrastive loss function is a pairwise loss function i.e. reduces the similarity between query and target $L_c(x_i, x_i^{\pm}) = d(x_i, x_i^{\pm})$; where d is the distance metric. However, triplet loss leverages on triplets (x_i, x_i^-, x_i^+) which should be carefully selected to utilize the benefit of the function $L_t(x_i, x_i^-, x_i^+) = d(x_i, x_i^+) - d(x_i, x_i^-) + \alpha$; where $d(x_i, x_i^+)$ and $d(x_i, x_i^-)$ are the distances between query and positive pair and query and negative pair respectively. Note that triplet and pair selection is an expensive process and the space complexity becomes exponential.

5.2.2 Normalization Techniques

To accelerate the training process of neural networks, normalization was introduced and is still a common operation in modern neural network models. Batch normalization [104] was proposed to speed up the training process by reducing the internal covariate shift of immediate features. Scaling and shifting the normalized values becomes necessary to avoid the limitation in representation. The normalization of a layer L can be defined as $\hat{L}^i = \frac{x^i - E[x^i]}{\sqrt{Var[x^i]}}$ where the layer L is normalized along the i-th dimension where $x = (x_1, x_2, ..., x_n)$ represents the input, $E[x^i]$ represents the mean of activation computed and $Var[x^i]$ represents the variance. The work in [105] shows that normalization aids convergence of the network. Recently, weight normalization [106] technique was introduced to normalize the weights of convolution layers to further speed up the convergence rate.

5.2.3 Hypersphere Embedding Techniques

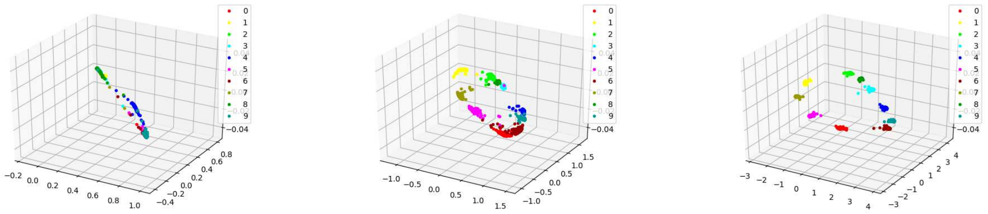
Different works in literature have explored different hyper-spaces for projection of learned features to figure out manifold with maximum separability between the deep features. Hypersphere embedding is one of the technique where the learned features are projected onto a hypersphere with the $L2$-normalize layer i.e. $\hat{x} = \frac{x}{\|x\|}$. Works in literature have employed hypersphere embedding for face recognition and verification tasks [107, 108, 109]. These techniques function by imposing discriminative constraints on a hypersphere manifold. As [104] explains that scale and shift is necessary to avoid the discriminatory limitations and are introduced as $y^{(i)} = \gamma^{(i)}\hat{x}^{(i)} + \beta^{(i)}$; where γ, β are learnable parameters. Inspired from this work, techniques such as [107] explore $L2$-normalize layer followed by scaling layer which scales the projected features by a factor α i.e. $\frac{\alpha x}{\|x\|}$ where α is the radius of the hypersphere and can be both learnable and predefined. However, in [107] the α is restricted to the radius of hypersphere and normalizes the features only. Furthermore, [109] normalizes the weights of last inner-product layer only and does not explore the scaling factor. The work presented in [108] optimizes both weights and features, and defines the normalization layer as $\|x\|_2 = \sqrt{\sum_i x_i^2 + \in}$ without exploring the scaling factor.

5.2.4 Revisiting Softmax-based Techniques

A generic pipeline for classification tasks consists of a CNN network learning the features of the input coupled with softmax as a supervision signal. We revisit the softmax function by looking at its definition $L_s = -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$; where $\mathbf{x}$ is the learned feature, $W_i \in \mathbb{R}$ denotes weights in the last fully connected layer and $b_i \in \mathbb{R}^n$ is the bias term corresponding to class i. It is clear that $W_i^T x_i + b_i$ is responsible for the class decision which forms intuition for the necessity of the fully connected layer after normalization. The work in [109] reformulates softmax and introduces an angular margin and modifies the decision boundary of softmax as $\|x\|(cosm\theta_1 - cos\theta_2) = 0$ for class 1 and $\|x\|(cos\theta_1 - cosm\theta_2) = 0$ for class 2. This differs from standard softmax in a sense that [109] requires $cos(m\theta_1) > cos(\theta_2)$ for the learned feature $\mathbf{x}$ to be correctly classified as class 1. This reformulation results in a hypersphere embedding due to the subtended angle. Similarly, [107] constraints the softmax by adding a normalization layer.

5.3 Proposed Method

In this section, we explore the intuition behind the inward scale layer and explain why normalization along with a fully connected layer is necessary before the softmax. We term a normalization layer along with the inward scaling factor as the inward scale layer. The reason behind this terminology is that normalization without the inward scaling acts as constraint imposer on the feature space and hampers the discriminative ability of the network. Furthermore, network struggles to converge if either of the layers are removed i.e. normalization, inward scale factor and fully connected. We set some terminology before proceeding with the explanation in Table 5.1.



(a) Plot at epoch 3.

(b) Plot at epoch 15.

(c) Plot at epoch 30.

Figure 5.2: Plots on test set of MNIST dataset at different epochs. The figure shows realistic plots of test set. At epoch 3 the projection of data points on hypersphere embedding space is in initial stages with little to no inward scaling. However, at epoch 15 effects of inward scaling are visible with the projection being maximum scaled at epoch 30. (Best viewed in color)

Table 5.1: Some important terminology used throughout this chapter.

Terminology	**Explanation**
M	Input manifold
$f(M)$	Projected hypersphere manifold
x_i	Learned features of class i
W_i	Weight of class i
b_i	Bias of class i
ξ	Inward scale factor
$IS(x, \xi)$	Inward scale layer with feature x and scale factor ξ
$FC(W, x, b)$	Fully connected layer with weight W, feature x and bias b

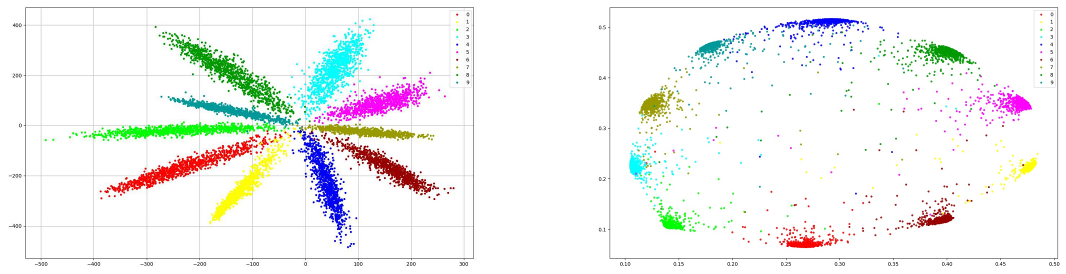
The work in [108] establishes that softmax function always encourages well-separated features to have bigger magnitudes resulting in radial distribution Figure 5.3a. However, the effect is minimized in Figure 5.3b because of the $IS(x, \xi)$.

5.3.1 Inward Scale Layer

In this paper, we define the inward scale layer as the normalization layer along with the inward scale factor ξ. The normalization layer can be defined as in Equation 5.1.

$$\hat{x} = \frac{x}{\|x + \mathcal{E}\|} \tag{5.1}$$

where $\mathcal{E}$ is the factor to avoid division by zero. Note that it is unlikely that norm $\|x\| = 0$, but to avoid the risk, we introduce the factor. Inspired from the works in literature [107, 106] we further introduce a scale factor ξ. Unlike employing it in the



(a) Plot on test set of MNIST reduced to 2-dimensional features with the softmax as supervision signal without the $IS(x, \xi)$.

(b) Plot on test set of MNIST reduced to 2-dimensional features with the softmax as supervision signal with the $IS(x, \xi)$.

Figure 5.3: Comparison of employing softmax with (a) and without (b) the inward scale layer. The softmax tends to have a radial distribution whereas with $IS(x, \xi)$ the distribution changes to hypersphere. Note that the plot (b) has some variation between the features in a radial fashion. This is due to the tendency of softmax. Note that figure (b) is slightly off from the ideal hypersphere embedding, since the features are extracted from the half trained network to establish analogy with the softmax, this scenarios takes place. (Best viewed in color)

product fashion as in [107], we couple with the norm in inverse fashion to ensure the scaling of the features as they are projected onto the manifold $f(M)$. In other words, we couple the factor ξ with $\|x\|$ to enhance the norm of the features instead of bounding entire layer. The Equation 5.2 is modified as $\hat{x} = \frac{x}{\xi(\|x+\mathcal{E}\|)}$. L2-norm can be re-written as $\|x\| = \sqrt{\sum_i x_i^2 + \mathcal{E}}$. Thus, $IS(x, \xi)$ can be formulated as follows.

$$\hat{x} = \frac{x}{\xi(\sqrt{\sum_i x_i^2 + \mathcal{E}})} \tag{5.2}$$

where x_i is the feature from the previous layer. Note that the factor ξ is not trainable. We experiment with different values of ξ and find that maximum separability is obtained with $\xi = 100$, see appendix A for experiments with different values of ξ.

The CNN layers are responsible for providing a meaningful feature space, without the $FC(W, x, b)$ layer, learning non-linear combinations of these features would not be possible. Simply put, the features are classified into different classes due to $FC(W, x, b)$ layers followed by a softmax layer. The Figure 5.3b in [107] visually illustrates the effect of L2-constrained softmax. On comparing it with our Figure 5.2c we visually see the effects of the inward scale layer. It is necessary to note that we do not modify the softmax and employ it as it is with the $IS(x, \xi)$ which in turn benefits the network with faster convergence and the learned features are discriminative enough for efficient classification and retrieval without the need for any metric learning. As the module is fully differentiable and is employed in end-to-end fashion, the gradient with respect to x_i is given as $\frac{\partial L}{\partial x_i}$ and can be solved using the chain-rule, see appendix B for the prove and appendix C for learning curves of the $IS(x, \xi)$.

5.4 Empirical Results

In order to quantify the effect of layer, in this section we report results of the $IS(x, \xi)$ layer on multiple datasets. We report results of different works available in the literature followed by the results of our work. Note that in order to demonstrate the modular nature of $IS(x, \xi)$ layer, we perform experiments with different baseline networks containing the proposed layer. The layer coupled architecture can be trained with standard gradient descent algorithms. In all of the following experiments we employ Adam [70] optimizer with an initial learning rate of $1e - 2$ and employ weight decay strategy to prevent indefinite growing of $\|x\|_2$ because after updating $\|x + \frac{\partial L}{\partial x}\| > \|x\|_2$ for all cases.

5.4.1 Classification Results

5.4.1.1 MNIST and FashionMNIST

For the basic experiment to quantify results of proposed layer, we perform the test on MNIST and FashionMNIST dataset which are famous benchmark dataset for neural networks. Table 5.2 demonstrates the results of $IS(x, \xi)$ layer and LeNet and compares it with available works in literature.

Table 5.2: Accuracy on MNIST and FashionMNIST test set in (%).

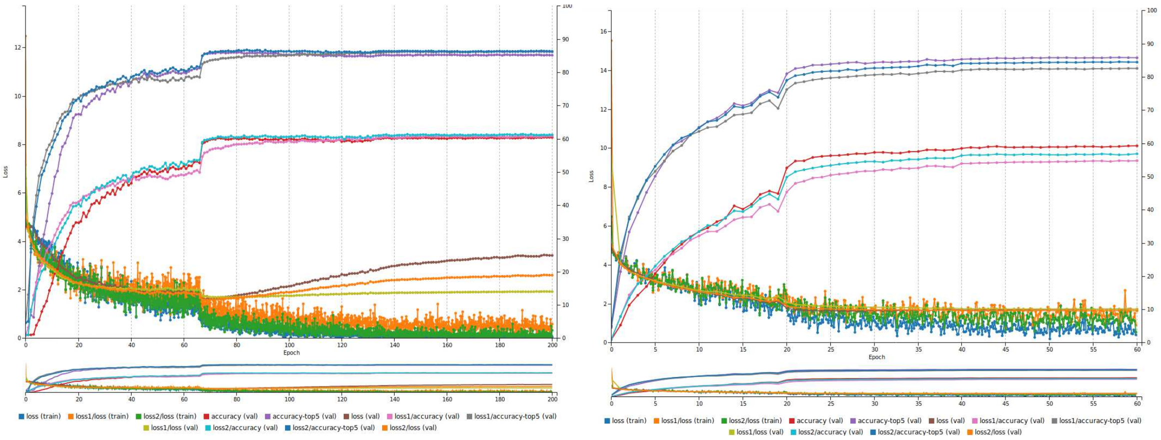
Methods	Dataset	**Accuracy** (%)
Softmax Loss	MNIST	98.64
Ours (without $IS(x, \xi)$)	MNIST	98.40
Ours (with $IS(x, \xi)$)	MNIST	99.33
Ours (without $IS(x, \xi)$)	FashionMNIST	89.64
Ours (with $IS(x, \xi)$)	FashionMNIST	93.00
Ranjan et. al [107]	MNIST	99.05
Zhong et. al [110]	FashionMNIST	96.35

5.4.1.2 SVHN

For the next experiment, we perform the test on SVHN dataset. Since MNIST and FashionMNIST are low resolution, grayscale and synthetic datasets, we test the layer on datasets with increasing complexity. Table 5.3 demonstrates the results of $IS(x,\xi)$ layer.

Table 5.3: Accuracy on SVHN test set in (%). LeNet is the baseline network for both the experiments.

Methods	Dataset	**Accuracy** (%)
Ours (Without $IS(x,\xi)$)	SVHN	93.20
Ours (With $IS(x,\xi)$)	SVHN	95.05
Zagoruyko et. al [111]	SVHN	98.46



(a) Training loss graph with the $IS(x,\xi)$ layer on CIFAR100 dataset using GoogleNet as baseline network. Classification accuracy is 60.44..

(b) Training loss graph without the $IS(x,\xi)$ layer on CIFAR100 dataset using GoogleNet as baseline network. Classification accuracy is 59.23..

Figure 5.4: Plots of training loss on CIFAR100 dataset with and without the proposed layer $IS(x,\xi)$ using GoogleNet as a baseline architecture with no pre or post processing.

5.4.1.3 CIFAR100

We perform an additional experiment on the CIFAR100 dataset to confirm the efficacy of the proposed layer. This experiment is particularly interesting because it augments an important claim behind the $IS(x,\xi)$ layer. We employ GoogleNet for this experiment for two reasons: (i) to verify that introduced layer can be coupled with GoogleNet and (ii) CIFAR100 is a large dataset compared to the datasets previously employed, thus, the accuracy with networks like LeNet is not satisfactory. Table 5.4 demonstrates the results of GoogleNet on CIFAR100 with and without the $IS(x,\xi)$ layer. Figure 5.4 visualizes the training graph with and without the proposed unit. It is interesting for readers to note the difference between the two graphs. Note that we imply the idea that projection and scaling happens during each pass and almost simultaneously due to scaling just before the projection. This is the major reason why loss behaves in variating fashion in the start. It should be noted that this does not mean the network struggles to converge.

Table 5.4: Accuracy on CIFAR100 dataset test set in (%). We employ GoogleNet for this experiment.

Methods	Dataset	**Accuracy** (%)
Ours (Without $IS(x, \xi)$)	CIFAR100	59.23
Ours (With $IS(x, \xi)$)	CIFAR100	60.44
Cireşan et. al [112]	CIFAR100	64.32
Goodfellow et. al [113]	CIFAR100	65.46
Springenberg et. al [114]	CIFAR100	66.29

5.4.2 Retrieval Results

In this section we report the retrieval results on FashionMNIST dataset. Most retrieval systems employ Recall@K as a metric to compute the scores. R@K is the percentage of queries in which the ground truth terms are one of the first K retrieved results. To retrieve results, we take query image and simply compute nearest neighbor (euclidean distance) between all images and sort results based on the distance. The first five distances correspond to Recall@K (K = 5) results and so on. We report results for $K = 1, 5, 10$. Since this is a unimodal retrieval, images are at the input and retrieval end. It is known that Recall@K increases even if one true positive out of Top K is encountered, so the results are almost similar. For a more valid quantitive analysis, we also present results of average occurrence of true positives (TP) in Top K. For retrieval, distance minimization is the major objective which softmax alone can not handle efficiently, thus we employ contrastive loss introduced by [101] along with softmax for the retrieval problem which shows that the proposed layer $IS(x, \xi)$ can function regardless of the architecture and loss function.

5.4.3 Result Discussion

We explore classification and retrieval tasks with and without the $IS(x, \xi)$ layer. The reported results indicate the superior performance of architecture with the $IS(x, \xi)$ layer. It is important to note that the each experiments is run 5 times and k-fold validation methodology is employed. The architecture with $IS(x, \xi)$ layer maintains the upper bound over its counter part without the $IS(x, \xi)$ layer. In Table 5.5, the **TP with**

Table 5.5: Recall@K and average occurrence of true positives (TP) in Top K scores for FashionMNIST test set with and without the $IS(x, \xi)$. Note that LeNet is the baseline architecture.

	Without $IS(x, \xi)$	**With** $IS(x, \xi)$	**TP with** $IS(x, \xi)$	**TP without** $IS(x, \xi)$
R@1	86.75	88.75	89.74	86.70
R@5	95.63	95.88	89.90	86.60
R@10	97.22	97.33	90.00	85.60

$IS(x, \xi)$ and **without** $IS(x, \xi)$ indicate the average occurrence of true positives in TopK retrieved results. The reason we employ this metric is Recall@K is incremented even if a single true positive is encountered out of TopK and thus the results with and without $IS(x, \xi)$ layer are almost similar. However, with average true positive, we compute the actual number of true positives out of TopK and compute the average to report more discriminative comparison between the two. Furthermore, we compare our work with state-of-the-art approaches. Note that most of the works obtaining state-of-the-art perform data preprocessing, while we do not employ any pre or post processing technique for any experiment performed and use single optimization policy without fine-tuning hyperparameters for any specific task.

5.5 Conclusion and Future Work

In this chapter, we proposed a novel $IS(x, \xi)$ layer for embedding the learned deep features into hypersphere. We propose that hypersphere embedding is important for discriminative analysis of the features. We verify the claim with extensive evaluation on multiple classification and retrieval tasks. Furthermore, the layer module can be added to any network and is fully differentiable and can be trained end-to-end with any network.

In future, we would like to explore different hyperspaces for discriminatively embedding feature representations. Furthermore, we would like to explore constraint-enforced hyperspaces where networks learns a mapping function under certain constraints thus resulting in a desired embedding.

6
Supplementary

6.1 Introduction

The aim of deep metric learning is to learn a similarity metric from data. The similarity metric can be used to compare or match new samples from previously unseen data. In recent years, deep metric learning has gained considerable popularity following the success in deep learning. Deep metric learning can be applied to numerous applications such as retrieval [99, 115], clustering [116], feature matching [117] and verification [103, 60]. Extreme classification [118, 119] with enormous number of classes can also take advantage of deep metric learning methods because of its ability to learn the general concept of distance metrics.

Typically, deep metric learning methods are built on state-of-the-art CNNs [22, 21, 120]. Deep metric learning methods produce an embedding of each input so that a certain loss, related to distance between two images, is minimized. In other words, embedding produced by metric learning methods are optimized to push examples of similar classes closer, conversely examples belonging to different classes are far from them. Such embedding is robust against intra-class variation which makes such methods suitable to learn similarity. Existing works take randomly sampled pairs of similar and dissimilar inputs or triplets consisting of query, positive and negative inputs to compute the loss on individual pairs or triplets.

Computer vision community has extensively used MNIST dataset in different applications including similarity. However, the dataset has only few seemingly similar classes, making it less effective for deep metric learning methods. In this chapter, a new handwritten dataset named Urdu-Characters is created in a similar way as MNIST dataset. Furthermore, we build Siamese and Triplet networks on Urdu-Characters and MNIST datasets to show that a Triplet network is more powerful than a Siamese network. We demonstrated that the performance of a Siamese or Triplet network can be improved further using most powerful underlying CNNs i.e. AlexNet [21] and GoogleNet [22].

6.2 Related Work

Kulis [121] provides a comprehensive survey on advances in metric learning. Siamese models have been used for very different purposes. For example Bromley *et al.* [122] presented a Siamese network for signature verification, while Chopra *et al.* [103] used a similar network for face verification. They pointed out a complete freedom in the choice of underlying architecture to build such family of networks. This observation is extremely important as future variants of Siamese network are built on top of more powerful architectures i.e. AlexNet [21], GoogleNet [22] etc.

With rise in e-commerce websites, deep metric learning methods are extensively employed in image retrieval applications, for example Bell and Bala [123] used variants of Siamese network to learn an embedding for visual search in an interior design context. The embedding produced by such network is then used to search for products in the same category, searching across categories and looking for a product in an interior scene. They concluded that using higher dimension of the embedding makes it easier to satisfy constraints in loss function. However, higher embedding dimension will significantly increase the amount of space and time required to search image in retrieval applications. We show that higher embedding dimension produces better results for such networks. However, the choice of embedding dimension is based on the application context. Veit *et al.* [115] extended the Siamese network to answer this question: *'What outfit goes well with this pair of shoes?'*. The proposed framework learns compatibility between items from different categories consisting of outfit and shoes. In other words, it goes beyond the notion of similarity using the notion of style. Their work is also one of the interesting applications of a Siamese network. Wang *et al.* [58] presented deep ranking model to learn fine-grained image similarity models based on triplet loss. Schroff *et al.* [60] used the similar loss for face verification, recognition and clustering. Authors also presented an online triplet mining method for creation of triplets. Similarly, we perform experiments with three triplet sampling strategies to analyze the impact of triplet creation on the network. We compare the performance of Siamese and Triplet networks with different underlying CNN architectures. We also analyzed the impact of embedding dimensionality on these deep metric learning methods. These two aspects of our work are not deeply explored in related works. Typically, deep metric learning methods used MNIST dataset in experiments. However, with the introduction of Urdu-Characters dataset, we provide researchers with a dataset with higher number of classes and ambiguities among classes. The nature of this dataset can be ideal for deep metric learning and classification tasks.

6.3 Deep Metric Learning Methods

A Siamese or Triplet network learns distance metric where similar examples are mapped close to each other and dissimilar examples are mapped farther apart.

6.3.1 Siamese Network

Siamese network (Figure 6.1a) is popular among tasks that involve finding similarity or a relationship between two comparable things. The network is characterized by using the contrastive loss function during the training which pulls together items of a similar class while pushing apart items of different classes. The formula is shown below:

$$\begin{aligned} L_s(x_i, x_i^{\pm}) = \sum_i^N & [(1-y)\|f(x_i) - f(x_i^{\pm})\|_2^2 \\ & +y \cdot max(0, \alpha - \|f(x_i) - f(x_i^{\pm})\|_2^2)] \end{aligned} \tag{6.1}$$

where N stands for the number of images in the batch, $f(\cdot)$ is the feature embedding output from the network, $\|f(x_i) - f(x_i^{\pm})\|_2^2$ is the Euclidean distance to measure the similarity of extracted features from two images, and the label $y \in \{0, 1\}$ indicates whether a pair $(x_i, x_i^{\pm})$ is from the same class or not.

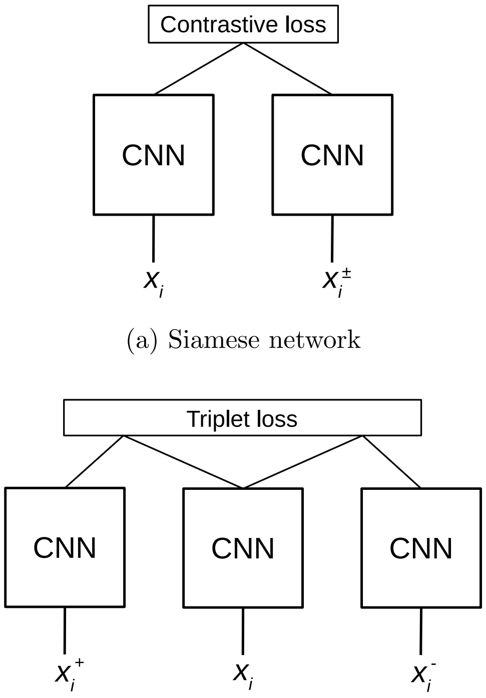
The training process for this kind of network is done feeding a pair of images x_i, $x_i^{\pm}$ and a label $y \in \{0, 1\}$ representing the similarity or dissimilarity between images.

6.3.2 Triplet network

The Triplet network (Figure 6.1b) is an extension of the Siamese network. It consists of three instances of the same feed-forward network (with shared parameters). The triplet loss [60] is trained on a series of triplets $\{x_i, x_i^+, x_i^-\}$, where x_i and x_i^+ are images from the same class, and x_i^- is from a different class, as reported in Equation 6.2. The triplet loss is formulated as following:

$$\begin{aligned} L_t(x_i, x_i^-, x_i^+) = \sum_i^N & max(0, [\|f(x_i) - f(x_i^+)\|_2^2 \\ & -\|f(x_i) - f(x_i^-)\|_2^2] + \alpha]) \end{aligned} \tag{6.2}$$

where $f(x_i), f(x_i^+), f(x_i^-)$ mean features of three input images and α is a margin that is enforced between positive and negative pairs.



(a) Siamese network

(b) Triplet network

Figure 6.1: Graphical representation of networks with Contrastive and Triplet loss functions used in this work.

6.3.3 Covolutional Neural Networks

A Siamese or Triplet network is built on top of underlying CNN architecture as shown in Figure 6.1. Typically, a CNN structure consists of various stages or layers such convolutional, pooling and rectification. Parameters in each layer are learned from training data to optimize performance on some tasks. AlexNet [21] is considered first CNN model successfully applied for image classification and starting from this architecture many new architectures have been presented in recent years. In this work, we use three well-known CNNs (LeNet, AlexNet and GoogleNet), as underlying architectures to build a Siamese network or Triplet network. LeNet has only two convolutional layers while AlexNet has 5 convolutional layers and GoogleNet has many more layers. It is important to note that 'softmax' layer is removed from these architectures to obtain D-dimensional embedding.

6.3.4 Triplet Sampling

We employ three different strategies for triplet creation in our experiments. We want to evaluate if the creation process has an impact on the overall performance. The first

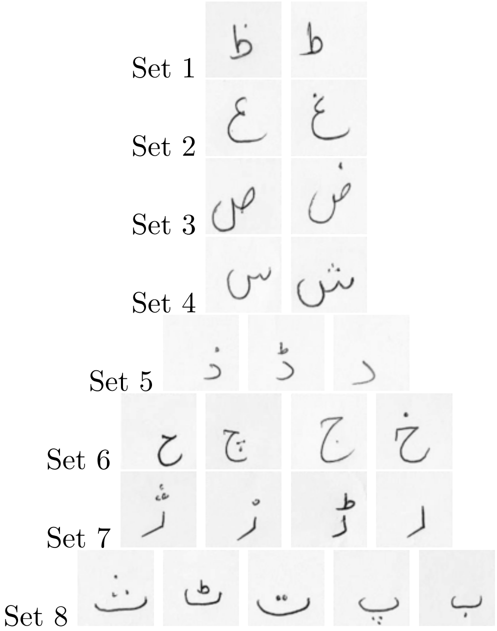


Figure 6.2: Interesting set of classes in Urdu-Characters dataset.

strategy that we employ consists of random selection of an image from the dataset, then we select one image belonging to the same class as positive sample and one belonging to a different class as negative sample. These images are selected randomly within these two sets.

The second strategy chooses a random image from the dataset, then extracts the most similar image of the same class and the most dissimilar from all other classes, excluding the class of the query image. To determine image similarity, we compute the Euclidean distance between them using feature vectors extracted from a CNN.

The third strategy differs from the second one on the selection of most similar image as positive image to the query image. This strategy selects the most dissimilar image in the same class as positive image. The vice-versa is for negative image. As reported in Equation 6.2, we expect to obtain best results with the third strategy because during the training process there would be higher error, making the backpropagation process more effective, while the second strategy would apply minimum adjustment within each step because of the similarity between query and positive image and the large difference between the query and the negative image.

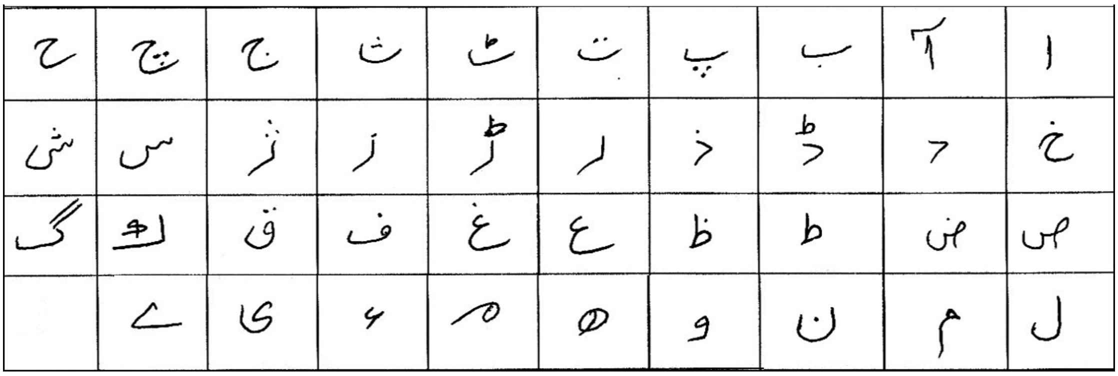


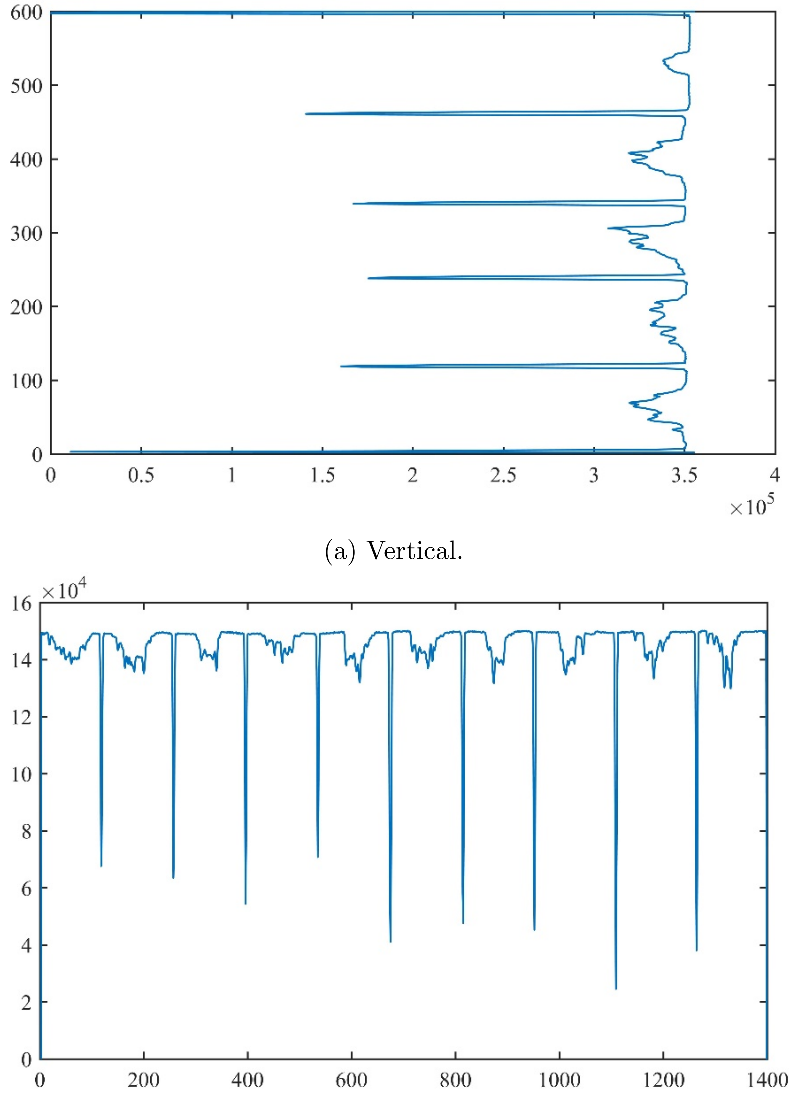
Figure 6.3: Typical handwritten response received from a student on a printed plain paper.

6.4 Dataset

The first dataset we use in experiments is the original MNIST [120] consisting of $60,000$ gray-scale images of handwritten digits $(0 - 9)$ and a corresponding set of $10,000$ test images with 28×28 pixels. MNIST dataset is extensively used in deep learning methods and is considered benchmark dataset. However, MNIST dataset has only few seemingly similar classes. This lead us to build a new handwritten dataset named Urdu-Characters, built in a similar way as MNIST dataset. The nature of characters in handwritten Urdu-Characters dataset is ideal for deep metric learning methods. There are some sets of classes available in Urdu-Character dataset which are seemingly similar however belong to different classes as shown in Figure 6.2.

Urdu-Characters dataset is collected on a printed plain paper with an $6in \times 2in$ box and 10×4 grid. To collect the data, a group of undergraduate students at University of Engineering and Technology, Lahore Pakistan participated in the activity. In particular, students are asked to write Urdu characters in a specific sequence from right to left. We received 560 responses from students. The collected forms are then scanned at 300 dots per inch resolution in 8 bit gray scale image for further processing. Figure 6.3 shows a response example of a student having written all Urdu characters from right to left.

The vertical and horizontal projections of student responses are obtained to detect grid lines for character segmentation. Figure 6.4 show these projections shows veritical and horizontal projections. Each projection is obtained summing all rows to first row and thus obtaining a plot. A similar procedure is used for horizontal projection. The projection lines with 85% or less sum were treated as separating lines and characters between them were separated. Extracted characters were normalized and converted into a 64×64 pixels image with 8 bit depth. Figure 6.5 shows some examples of extracted



(a) Vertical.

(b) Horizontal

Figure 6.4: Vertical and horizontal projections to detect vertical and horizontal grid lines of handwritten response received from a student.

handwritten characters. There are $20,324$ segmented characters grouped in 39 classes with $15,251$ characters for train and $5,073$ characters for test set.

6.5 Experiments

We compare the performance of a Siamese and Triplet network on MNIST and Urdu-Characters datasets. In addition, we want to compare the performance of Siamese and

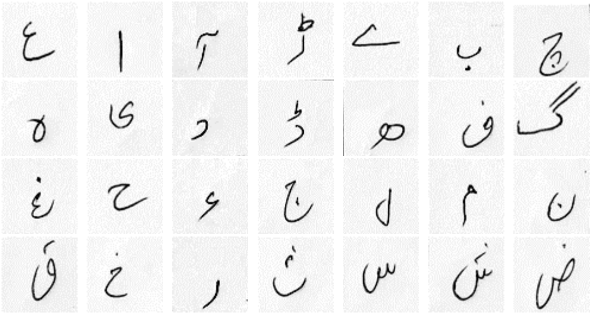


Figure 6.5: Extracted Urdu characters from a student response on a printed plain paper.

Triplet Networks built on top of different underlying CNNs architectures with different embedding dimensionality. To achieve these objectives, we performed a series of experiments on both datasets as follow:

– Build a Siamese network

– Compare the performance of Siamese network built on top of different underlying CNNs (LeNet and AlexNet)

– Build a Triplet network

– Compare the performance of a Triplet and Siamese network

– Compare the performance of Triplet networks built on top of different underlying CNNs architectures

We use Caffe [124] and The NVIDIA Deep Learning GPU Training System deep learning frameworks, which contains efficient GPU implementations for training CNNs. In experiments, accuracy is employ to measure the performance of different network settings. Table 6.1 shows a Siamese network settings for MNIST and Urdu-Character datasets built on top of LeNet. We use $100,000$ pairs of similar and dissimilar randomly selected images to built a Siamese network. Output embedding dimensionality of the network is 256. The last column in Table 6.1 shows the accuracy obtained by Siamese network. The accuracy value of a Siamese network shows that network built on top of LeNet does not perform well on a complex dataset like Urdu-Characters. This leads us to built a Siamese network on top of more powerful network i.e. AlexNet for Urdu-Characters dataset. Table 6.2 shows that a Siamese network built on top of AlexNet produces better results compared to the same model built on top of LeNet.

These results lead us to built a competitor of Siamese network i.e. a Triplet network. We use $100,000$ triplets consisting of query, positive and negative images to build a Triplet network. Table 6.3 shows a Triplet network settings for MNIST and Urdu-Character datasets built on top of LeNet. Accuracy values of both datasets for a Triplet

Table 6.1: Siamese network settings built on top of LeNet for MNIST and Urdu-Characters datasets.

Dataset	Resolution	Network	Embedding	Pairs	Accuracy
MNIST	28×28	LeNet	256	$100,000$	96.23
Urdu-Character	64×64	LeNet	256	$100,000$	27.79

Table 6.2: Siamese network settings built on top of LeNet and AlexNet for Urdu-Characters.

Dataset	Resolution	Net.	Emb.	Pairs	Accuracy
Urdu-Character	64×64	LeNet	256	$100,000$	27.79
Urdu-Character	64×64	AlexNet	256	$100,000$	61.46

Network is higher than accuracy values of a Siamese Network as shown in Figure 6.6. This proves that a Triplet Network is more powerful than a Siamese Network. We also built a Triplet network on top of AlexNet to obtain better results than a Triplet network built on top of LeNet. Table 6.4 shows the accuracy values of a triplet network built on top of LeNet and AlexNet. This leads us to built a Triplet network on even more powerful network like GoogleNet. However, to perform this experiment we need an image size of 256×256, hence, we up sample Urdu-Characters dataset images. Accuracy values in Table 6.5 show that a Triplet network built on top of GoogleNet is more powerful than a network built on top of AlexNet.

We compare the impact of the embedding dimensionality on Siamese and Triplet networks. This leads us to built a Triplet network on top of AlexNet and GoogleNet and a Siamese network on top of LeNet and AlexNet with $128, 256, 512$ embedding dimensionality as shown in Figure 6.8 and 6.7a. These results show that higher embedding dimensionality obtain better accuracy values. However, the choice of embedding dimen-

Table 6.3: Triplet network settings built on top of LeNet for MNIST and Urdu-Characters datasets.

Dataset	Resolution	Net.	Emb.	Triplets	Accuracy
MNIST	28×28	LeNet	256	$100,000$	98.23
Urdu-Characters	64×64	LeNet	256	$100,000$	53.45

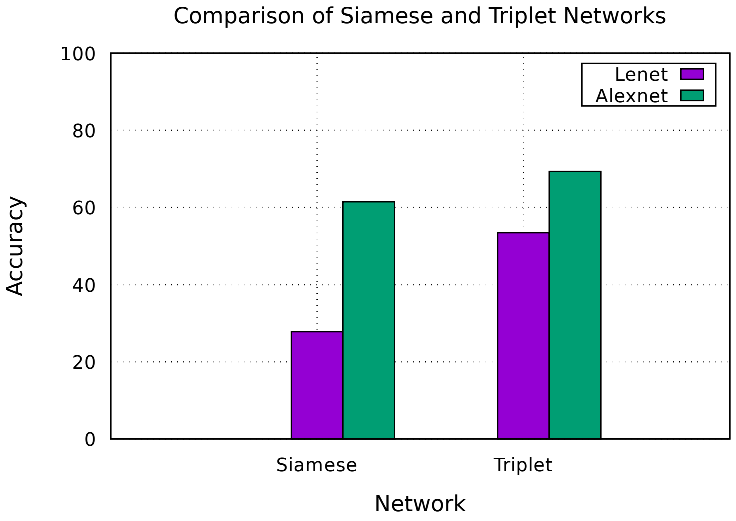


Figure 6.6: [
Comparison of Triplet and Siamese networks Built on Top of LeNet and AlexNet]Comparison of Triplet and Siamese networks built on top of LeNet and AlexNet with 256 embedding dimensionality for Urdu-Characters dataset. We employed network settings mentioned in Table 6.2 and Table 6.4 to built a Siamese and triplet network respectively.

sionality depends considerably on the application context. For example, using search by example system, a higher embedding dimensionality could make the process very slow.

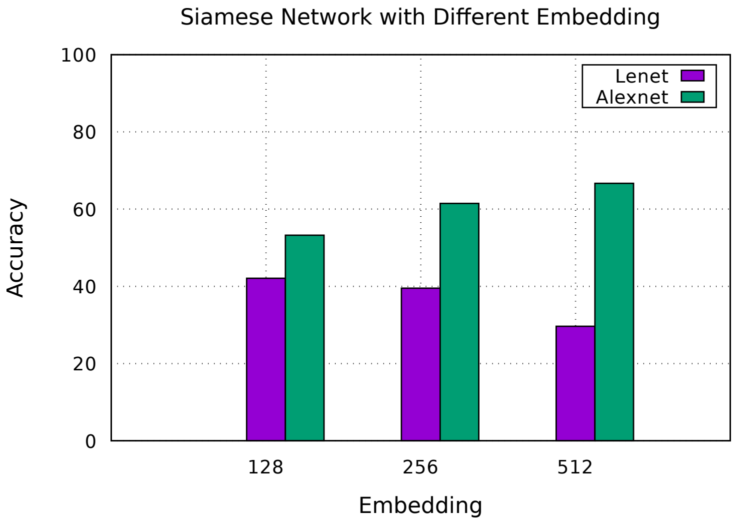
Finally, we compare the effect of different triplet sampling strategies on the performance of the triplet network. Table 6.6 shows the performance of three sampling strategies discussed in section 6.3.4. Results of these strategies are comparable however, strategy 3 is better than other two strategies because it violates the triplet constraints. However, we believe that the triplet selection strategy depends on the variation in the dataset. In our Urdu-Characters dataset, we have do not have high variability inside classes.

Table 6.4: Triplet network settings built on top of LeNet and AlexNet for Urdu-Characters dataset.

Dataset	Resolution	Net.	Emb.	Triplet	Accuracy
Urdu-Characters	64×64	LeNet	256	$100,000$	53.45
Urdu-Characters	64×64	AlexNet	256	$100,000$	69.35

Table 6.5: Triplet Network Settings Built on Top of AlexNet and GoogleNet for Urdu-Characters dataset.

Dataset	Resolution	Net.	Emb.	Triplet	Accuracy
Urdu-Characters	256 × 256	AlexNet	256	100, 000	69.98
Urdu-Characters	256 × 256	GoogleNet	256	100, 000	77.06



(a)]

Siamese network built on top of LeNet and AlexNet with 128, 256, 512 embedding dimensionalities for Urdu-Characters dataset. We employed same network settings mentioned in Table 6.2 to built the network.

Table 6.6: Comparison of triplet sampling strategies. We built triplet network on top of AlexNet with 15, 251 triplets for training and 5, 073 triplets for test. Embedding dimensionality for these networks is 128.

Strategy	Accuracy
Strategy # 1	59.88
Strategy # 2	61.48
Strategy # 3	61.78

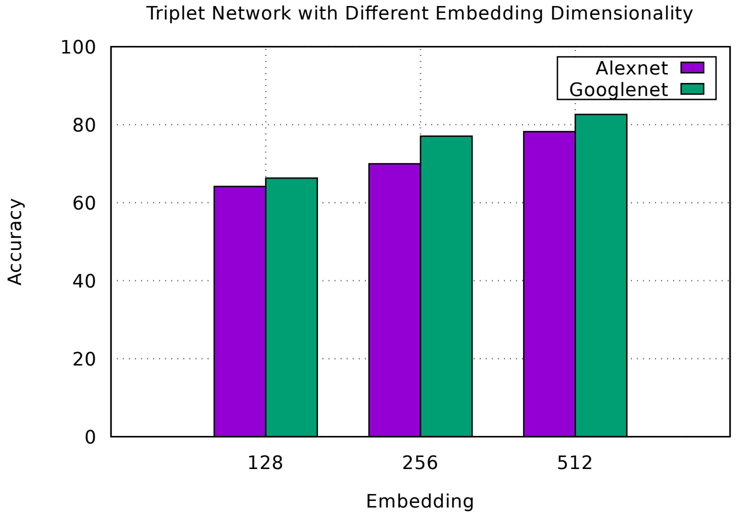


Figure 6.8: Triplet network built on top of AlexNet and GoogleNet with $128, 256, 512$ embedding dimensionality for Urdu-Characters dataset. We employ the same network settings mentioned in Table 6.5 to built the network.

6.6 Conclusions

We built a handwritten Urdu-Characters dataset containing some sets of classes suitable for deep metric learning methods. We showed that Siamese network built on top LeNet performed well on MNIST dataset, but it did not reach good results on Urdu-Characters dataset, however, a Siamese network built on top of AlexNet obtains significantly better results on Urdu-Characters. A similar phenomenon also happened for a Triplet network, where the difference between using different underlying CNN architectures such as LeNet, AlexNet or GoogleNet is considerable in terms of overall accuracy. Furthermore, we compared three sampling strategies to create triplets to built a Triplet network, but we obtained comparable results. Usually, the use of different sampling strategies lead to different accuracy values due to the variation in the dataset, however not in our case with Urdu-Character dataset due to the fact that it does not have high variability inside classes.

7 Conclusions and Future Research Directions

7.1 Conclusion

In this doctoral thesis, we presented techniques to enhance representations from individual and multiple modalities for multimodal applications including classification, cross-modal retrieval, matching and verification on various benchmark datasets.

Recent years have seen an explosion in multimodal data on the web. It is therefore important to perform multimodal learning to understand the web. However, it is challenging to join various modalities because each modality has different representation and correlational structure. Therefore, we focus on improving representations from individual modalities to enhance multimodal representation and learning. Main contributions are listed below.

- “**Deep Latent Space Representations**” framework consisting of a single stream, end-to-end trainable network with a novel training procedure to map multiple modalities to shared latent space without pairwise or triplet information at the input.
- “**Deep Fused Representations Framework** for multimodal classification”
- A “**Visual Word Embedding Scheme**” that transforms Word2Vec word embedding into visual space. The scheme enhances the text representation.
- An “**Inwardly Scale Feature Representations**” to render similar instances closer and dissimilar instances distant. The approach improves the discriminative representation of image.

- "**Metric learning approaches**" to improve the image representations.

7.2 Perspectives for Future Research

This doctoral thesis proposed various contributions to improve representations from individual and multiple modalities on various multimodal applications such as classification, cross-modal retrieval and verification. These contributions are in no way complete solutions, and could be improved in several manners. In the following, we propose potential directions that can be explored further.

- Inverted image representations visualization methods are useful to analyze the representations from various convolutional layers [33, 125]. Interestingly, inverted representations provide several insights into the properties of the feature representation learned by the network. It would be interesting to visualize the inverted representations from the joint layer of a multimodal approach. The visualizations may provide information on the influence of one modality on the other at the combined layer for various multimodal applications.
- Last year seen an increased interest to transform text centric Named Entity Recognition problem into multimodal approach [14, 15, 16]. It would be interesting to utilize the representations discussed in Chapter 3 and Chapter 4 for the multimodal Named Entity Recognition task.
- "**Deep Latent Space Representations**" approach for Audio-Image is evaluated on Voxceleb dataset. The audio samples in Voxceleb dataset are in English language. It would be useful to extend Voxceleb dataset for other languages. The above approach can then be extended for multilingual cross-modal verification.

Colophon

- This thesis was written using LaTeX.
- The LaTeX template for this thesis was made by Carullo Moreno.
- Algorithms presented were developed using the Google TensorFlow[1], Pandas[2] and Scikit-learn libraries[3].
- All experiments have been run on a machine equipped with an Intel Core i7-6800K 3.50 GHz with 64 GB of RAM, three NVIDIA GTX 1080 and Linux Mint 19 Tara.

[1] https://www.tensorflow.org

[2] https://pandas.pydata.org

[3] https://scikit-learn.org

Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] C. A. Bhatt and M. S. Kankanhalli, “Multimedia data mining: state of the art and challenges,” *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 35–76, 2011.

[3] N. Srivastava and R. R. Salakhutdinov, “Multimodal learning with deep boltzmann machines,” in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[4] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González, “Gated multimodal units for information fusion,” *arXiv preprint arXiv:1702.01992*, 2017.

[5] D. Kiela, E. Grave, A. Joulin, and T. Mikolov, “Efficient large-scale multi-modal classification,” *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 5198–5204, 2018.

[6] I. Gallo, A. Calefati, and S. Nawaz, “Multimodal classification fusion in real-world scenarios,” in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 5. IEEE, 2017, pp. 36–41.

[7] 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*, 2016, pp. 5005–5013.

[8] D. Kiela and L. Bottou, “Learning image embeddings using convolutional neural networks for improved multi-modal semantics,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 36–45.

[9] C. W. Leong and R. Mihalcea, “Going beyond text: A hybrid image-text approach for measuring word relatedness,” in *International Joint Conference on Natural Language Processing*, 2011, pp. 1403–1407.

[10] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *Proceedings of Empirical Methods in Natural Language Processing, EMNLP 2016*, pp. 457–468, 2016.

[11] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, vol. 3, no. 5, 2018, p. 6.

[12] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[13] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[14] Q. Zhang, J. Fu, X. Liu, and X. Huang, "Adaptive co-attention network for named entity recognition in tweets," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] S. Moon, L. Neves, and V. Carvalho, "Multimodal named entity recognition for short social media posts," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers.* Association for Computational Linguistics, 2018, pp. 852–860.

[16] D. Lu, L. Neves, V. Carvalho, N. Zhang, and H. Ji, "Visual attention model for name tagging in multimodal social media," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1990–1999.

[17] O. Arshad, I. Gallo, S. Nawaz, and A. Calefati, "Aiding intra-text representations with visual context for multimodal named entity recognition," *arXiv preprint arXiv:1904.01356*, 2019.

[18] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[19] Y. Li, S. Wang, Q. Tian, and X. Ding, "A survey of recent advances in visual feature detection," *Neurocomputing*, vol. 149, pp. 736–751, 2015.

[20] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning.” 2017.

[24] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Advances in neural information processing systems*, 2015, pp. 649–657.

[25] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, “Very deep convolutional networks for text classification,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, vol. 1, 2017, pp. 1107–1116.

[26] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, October 2014, pp. 1746–1751.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, ser. NIPS’13, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

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

[29] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *International Conference on Machine Learning*, 2014, pp. 1188–1196.

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

[31] 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*, 2016, pp. 770–778.

[32] Y. Xiao and K. Cho, “Efficient character-level document classification by combining convolution and recurrent layers,” *arXiv preprint arXiv:1602.00367*, 2016.

[33] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5188–5196.

[34] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal processing magazine*, vol. 29, 2012.

[35] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[37] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, “Multimodal deep learning,” in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[38] D. Wang, P. Cui, M. Ou, and W. Zhu, “Deep multimodal hashing with orthogonal regularization,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[39] Y. Liu, L. Liu, Y. Guo, and M. S. Lew, “Learning visual and textual representations for multimodal matching and classification,” *Pattern Recognition*, vol. 84, pp. 51–67, 2018.

[40] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, “Vse++: Improving visual-semantic embeddings with hard negatives,” *arXiv preprint arXiv:1707.05612*, 2017.

[41] A. Nagrani, S. Albanie, and A. Zisserman, “Seeing voices and hearing faces: Cross-modal biometric matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8427–8436.

[42] ——, "Learnable pins: Cross-modal embeddings for person identity," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 71–88.

[43] R. Qiao, L. Liu, C. Shen, and A. v. d. Hengel, "Visually aligned word embeddings for improving zero-shot learning," *arXiv preprint arXiv:1707.05427*, 2017.

[44] Y. Zhang, B. Gong, and M. Shah, "Fast zero-shot image tagging," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[45] S. Dey, A. Dutta, S. K. Ghosh, E. Valveny, J. Lladós, and U. Pal, "Learning cross-modal deep embeddings for multi-object image retrieval using text and sketch," *arXiv preprint arXiv:1804.10819*, 2018.

[46] 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, 2018.

[47] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, and Y.-D. Shen, "Dual-path convolutional image-text embedding," *arXiv preprint arXiv:1711.05535*, 2017.

[48] C. Kim, H. V. Shin, T.-H. Oh, A. Kaspar, M. Elgharib, and W. Matusik, "On learning associations of faces and voices," *arXiv preprint arXiv:1805.05553*, 2018.

[49] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision.* Springer, 2016, pp. 499–515.

[50] 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.

[51] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik, "Improving image-sentence embeddings using large weakly annotated photo collections," in *European Conference on Computer Vision.* Springer, 2014, pp. 529–545.

[52] B. Klein, G. Lev, G. Sadeh, and L. Wolf, "Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[53] ——, "Associating neural word embeddings with deep image representations using fisher vectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4437–4446.

[54] J. Hu, J. Lu, and Y.-P. Tan, “Discriminative deep metric learning for face verification in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1875–1882.

[55] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, “Metric learning for large scale image classification: Generalizing to new classes at near-zero cost,” in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 488–501.

[56] B. Shaw, B. Huang, and T. Jebara, “Learning a distance metric from a network,” in *Advances in Neural Information Processing Systems*, 2011, pp. 1899–1907.

[57] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.

[58] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, “Learning fine-grained image similarity with deep ranking,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[59] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, “Matchnet: Unifying feature and metric learning for patch-based matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3279–3286.

[60] 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*, 2015, pp. 815–823.

[61] J. Weston, S. Bengio, and N. Usunier, “Wsabie: Scaling up to large vocabulary image annotation,” in *International Joint Conference on Artificial Intelligence*, 2011.

[62] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov *et al.*, “Devise: A deep visual-semantic embedding model,” in *Advances in neural information processing systems*, 2013, pp. 2121–2129.

[63] J. Lei Ba, K. Swersky, S. Fidler *et al.*, “Predicting deep zero-shot convolutional neural networks using textual descriptions,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4247–4255.

[64] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, “Grounding of textual phrases in images by reconstruction,” in *European Conference on Computer Vision*. Springer, 2016, pp. 817–834.

[65] A. Jabri, A. Joulin, and L. van der Maaten, "Revisiting visual question answering baselines," in *European conference on computer vision.* Springer, 2016, pp. 727–739.

[66] I. Gallo, S. Nawaz, and A. Calefati, "Semantic text encoding for text classification using convolutional neural networks," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 5. IEEE, 2017, pp. 16–21.

[67] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *INTERSPEECH*, 2017.

[68] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[69] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," *arXiv preprint arXiv:1511.06361*, 2015.

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2015.

[71] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, "Hierarchical multimodal lstm for dense visual-semantic embedding," in *Computer Vision (ICCV), 2017 IEEE International Conference on.* IEEE, 2017, pp. 1899–1907.

[72] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," *International Conference on Learning Representations*, 2015.

[73] L. Ma, Z. Lu, L. Shang, and H. Li, "Multimodal convolutional neural networks for matching image and sentence," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2623–2631.

[74] G. Park and W. Im, "Image-text multi-modal representation learning by adversarial backpropagation," *arXiv preprint arXiv:1612.08354*, 2016.

[75] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[76] S. Nawaz, M. K. Janjua, A. Calefati, and I. Gallo, "Revisiting cross modal retrieval," *arXiv preprint arXiv:1807.07364*, 2018.

[77] I. Gallo, A. Calefati, S. Nawaz, and M. K. Janjua, "Image and encoded text fusion for multi-modal classification," *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2018.

[78] D. Kiela and L. Bottou, "Learning image embeddings using convolutional neural networks for improved multi-modal semantics," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, October 2014, pp. 36–45.

[79] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[80] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[81] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia systems*, vol. 16, no. 6, pp. 345–379, 2010.

[82] E. Bruni, G. B. Tran, and M. Baroni, "Distributional semantics from text and images," in *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, ser. GEMS '11, 2011, pp. 22–32.

[83] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2539–2544.

[84] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, "Recipe recognition with large multimodal food dataset," *2015 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 1–6, 2015.

[85] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2016, pp. 507–517.

[86] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[87] A. Calefati, M. K. Janjua, S. Nawaz, and I. Gallo, "Git loss for deep face recognition," *British Machine Vision Conference*, 2018.

[88] J. Deng, Y. Zhou, and S. Zafeiriou, “Marginal loss for deep face recognition,” in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPRW), Faces “in-the-wild” Workshop/Challenge*, vol. 4, no. 6, 2017.

[89] K. Jarrett, K. Kavukcuoglu, Y. LeCun *et al.*, “What is the best multi-stage architecture for object recognition?” in *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 2146–2153.

[90] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *Advances in neural information processing systems*, 2014, pp. 1988–1996.

[91] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, “On the surprising behavior of distance metrics in high dimensional space,” in *International conference on database theory.* Springer, 2001, pp. 420–434.

[92] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is “nearest neighbor” meaningful?” in *International conference on database theory.* Springer, 1999, pp. 217–235.

[93] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[94] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Citeseer, Tech. Rep., 2009.

[95] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning.”

[96] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[97] Y. Ying and P. Li, “Distance metric learning with eigenvalue optimization,” *Journal of Machine Learning Research*, vol. 13, no. Jan, pp. 1–26, 2012.

[98] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, “Large scale metric learning from equivalence constraints,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 2288–2295.

[99] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep metric learning via lifted structured feature embedding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4004–4012.

[100] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, “Multi-manifold deep metric learning for image set classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1137–1145.

[101] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2006, pp. 1735–1742.

[102] N. A. I. G. S. Nawaz, A. Calefati, “Hand written characters recognition via deep metric learning,” in *13th IAPR International Workshop on Document Analysis Systems (DAS)*, vol. 05, 2018, pp. 417–422.

[103] S. Chopra, R. Hadsell, Y. LeCun *et al.*, “Learning a similarity metric discriminatively, with application to face verification,” in *CVPR (1)*, 2005, pp. 539–546.

[104] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[105] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[106] T. Salimans and D. P. Kingma, “Weight normalization: A simple reparameterization to accelerate training of deep neural networks,” in *Advances in Neural Information Processing Systems*, 2016, pp. 901–909.

[107] R. Ranjan, C. D. Castillo, and R. Chellappa, “L2-constrained softmax loss for discriminative face verification,” *arXiv preprint arXiv:1703.09507*, 2017.

[108] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, “Normface: l2 hypersphere embedding for face verification,” in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1041–1049.

[109] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphereface: Deep hypersphere embedding for face recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[110] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” *arXiv preprint arXiv:1708.04896*, 2017.

[111] S. Zagoruyko and N. Komodakis, “Wide residual networks,” *arXiv preprint arXiv:1605.07146*, 2016.

[112] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “High-performance neural networks for visual object classification,” *arXiv preprint arXiv:1102.0183*, 2011.

[113] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.

[114] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *International Conference on Leanring Representations*, 2015.

[115] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, "Learning visual clothing style with heterogeneous dyadic co-occurrences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4642–4650.

[116] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2016, pp. 31–35.

[117] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," in *Advances in Neural Information Processing Systems*, 2016, pp. 2414–2422.

[118] A. Choromanska, A. Agarwal, and J. Langford, "Extreme multi class classification," in *NIPS Workshop: eXtreme Classification, submitted*, 2013.

[119] Y. Prabhu and M. Varma, "Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2014, pp. 263–272.

[120] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[121] B. Kulis *et al.*, "Metric learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.

[122] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[123] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 98, 2015.

[124] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia.* ACM, 2014, pp. 675–678.

[125] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4829–4837.