University of Insubria
Department of Theoretical and Applied Sciences (DiSTA)

PhD Thesis in Computer Science
XXXII Cycle of Study

Discriminative Feature Learning for Multimodal Classification

Candidate:
ALESSANDRO CALEFATI

Thesis Advisor:
Prof. IGNAZIO GALLO

9th October 2019

To my Family

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my advisor Prof. Ignazio Gallo that allowed me to live this very special experience. He taught me to work hard even when things seem not to go in the right direction or good results look are not yet to come. This is the most important teaching I learnt and that I will take with me in my life.

I want to thank my colleague Shah Nawaz with which I collaborated during this three-years journey. It has been difficult but thanks to its support and its huge network of people, he helped me in getting out from difficulties. Thanks to him I knew some very good students who helped us in research projects and writing papers.

I gratefully acknowledge Prof. Pierluigi Gallo (Università degli Studi di Palermo) and Dr. Muhammad Haroon Yousaf (University of Engineering and Technology, Taxila Pakistan) for reviewing my thesis and for giving me precious and priceless comments.

Sincere thanks to 7Pixel that made my Ph.D. work possible, providing data which have been extensively used in the papers presented in this thesis. Moreover they supplied interesting cues and ideas, places and funds for our research activities, making our work even more exciting.

I would also like to thank actual and past members of the Applied Recognition Technology Laboratory (ArteLab): Alessandro Zamberletti, Lucia Noce and Riccardo La Grassa. A big thanks to my 7Pixel's colleagues, they have been a source of support and a good chance to improve my development skills.

I thank my Family that allowed me to reach this goal and for the support they offered during these years. They made me the person I am and I will never stop to be grateful for everything they did for me.

Last but not least, I express my gratitude to Daniela, she put up with me in hard moments of this experience and provided precious advice to keep working and to face problems I met.

Alessandro Calefati
Varese, 9th October 2019

1
Introduction

The purpose of this thesis is to tackle two related topics: multimodal classification and objective functions to improve discriminative power of features. First, I worked on image and text classification tasks and performed many experiments to show the effectiveness of different approaches available in literature, which are described in Chapter 3. Then, I introduced a novel methodology which can classify multimodal documents using single-modal classifiers merging textual and visual information into images and a novel loss function to improve separability between samples of a dataset.

The growth of the web and social networks have lead to an explosion of data, which often is characterized by a multimodal nature. For instance, on social networks people share posts using text and images or videos, which typically contain visual and audio information. Another effective example of multimodal information are marketplaces: in fact, on e-commerce websites online sellers advertise products using a combination of text and images to describe and show features of an item. Often, very different products on sale seem to be very similar if considering only the textual description or the image. Classification in this scenario is even more difficult taking into account the fact that data, typically, is affected by noise. In fact, often, it happens that the textual description of an item is very generic, unclear or even completely unrelated to the object it should describe, making the classification task even harder.

Tackling the text classification in social networks field is a challenging task, because posts on social platforms are characterized by short text, slang and irony or sarcasm. These characteristics of text are even more emphasized when talking about specific topics, such as politics.

All abovementioned observations lead to the fact that considering solely single modalities can lead to uncertain and poor classification results [1, 2], while the joint usage of text and images may help to overcome this problem. In this thesis, the importance of methods that can deal with such modality of data is shown, highlighting improvements obtained using multimodal classification in terms of performance and robustness against the lack of information or inconsistencies within data. Using multimodal classification, these issues, which can be considered as noise of a dataset, can be resolved.

Multimodal classification has been studied in literature and applied to many fields, ranging from computer science to health. The analysis of images and text on e-commerce websites, posts from social network with images, audios and text [3] or even analysis of biometric data [4, 5] to detect diseases are few examples.

Classification tasks have been tackled from decades obtaining very good results with traditional models like Support Vector Machine, decision trees, random forests and multi-layers perceptrons [6]. With the recent introduction of Convolutional Neural Networks (CNNs), results have been further improved making CNNs to become the *de-facto* standard in image classification. In addition to the strength of novel models, dealing with data coming from various sources is important, because, considering a single modality, often, is not enough to capture relevant information or semantic relationships.

Typically, multimodal classification has been performed with *ensemble-of-classifiers* approaches. Methods consist in the joint usage of multiple classifiers, which, then, are merged together to perform the final classification [7]. Boosting and Bagging methods achieved good results and opened the way to multimodal approaches [8]. The intuition behind the usage of multiple models is to create a more powerful and robust classifier to overcome errors which could be done during the prediction phase by single classifiers. For instance, ideally, when a model misclassifies a pattern, there should be another one performing a correct classification. Assigning weights to each classifier is a simple but effective method to reach this objective.

We tackled the multimodal classification topic, starting from simpler approaches such as Early Fusion and Late Fusion to more complex approaches based on word2vec (w2v) word embedding and CNNs. In the first part of this thesis, an approach which is able to perform multimodal classification merging information coming from images and text inside an image is presented. The advantage of this method is that it exploits capabilities of CNNs designed for image classification only to multimodal classification tasks, without making any changes to the architecture.

Approaches for classification working with multiple information sources lead to better performance overcoming errors induced by noisy datasets or incomplete data [9].

In addition to multimodal approaches, separability between classes plays a crucial role in improving classification performance. Datasets characterized by data of different classes being very close each other make the classification task very complex and error-prone; even state-of-the-art classifiers on these datasets obtain low performances. This scenario is even more challenging when introducing a method which represents multimodal features using only a single modality. This to be able to perform classification using a unique classifier without the need of applying changes. For example, in this thesis, it is presented an approach which uses CNNs, designed for image classification, for multimodal classification.

In an ideal scenario, we would like to have compact features of a class while very far features of other classes, in other words, having low intra-class variations and high inter-class distances. This is the objective of the second main work presented in this thesis.

Deep Metric Learning has been studied from decades. In fact, in literature there are many approaches trying to optimize the similarity or relative similarity between dataset samples [10, 11, 12, 13]. Most of them require the creation of a new datasets with a huge cardinality, made up of pairs or triplets, to train models effectively. In literature, majority of the works are performed on face verification and recognition tasks [14, 15, 16, 17, 18, 19, 20], however, discriminative feature learning is a recurrent task and not limited only to face classification topic. The second objective of this thesis is to propose a novel loss function to improve the discriminative power of a model, while avoiding the dramatic data expansion typical of previously cited approaches.

In the last part of this thesis, a novel loss function which minimizes intra-class and, at the same time, improves inter-class distances is proposed. This scenario increases results of classification, as shown by results obtained in [21].

2 Motivations and Background

In this Chapter, I review the literature related to main works of this thesis and will explain some ground concepts useful to understand the proposed approaches, explained in Sections 4 and 5.
In particular, in Section 2.1, most of previous approaches dealing with multimodal document classification, starting from works employing traditional classifiers to more recent approaches using CNNs are reported. There will be an analysis on pros and cons for each approach and motivation of taken choices. In Section 2.2, fundamental concepts of CNNs are explained. These notions are useful to understand our proposed approaches and advantages of using this kind of architectures to perform classification task. In Section 2.3 there is a brief summary of the work proposed in "Text understanding from scratch" [22] which has been used as inspiring model for the work described in Section 4. Datasets used in the evaluation process of main works are described in Section 2.4.

2.1 Motivations

Multimodal classification has been studied and applied to many topics, such as smart homes [23], surface classification from satellite images [24] and health [4] to name a few. In e-commerce, advertising companies are using preferences of users to select and show most suitable products for a specific target audience, or still, the joint analysis of images and audios, can be exploited for cross-modal biometric matching [25]. In the literature there are approaches dealing with traditional single-modal classification on text [26, 27] and images, but most of them solve the task with traditional classifiers like SVM [28], decision trees [29] and random forests [30].

I review the literature first considering the image classification task, then approaches dealing with text only and, finally, I will discuss about methods working on the combination of images and text for classification.

With the advent of the CNNs [31] and their stunning results, the task of image classification has changed drastically. Traditionally, the pipeline for image classification was made up by feature extraction performed, typically, by Histogram of Gradients (HOG) [32], Pyramid Histogram of Gradients (P-HOG) [33] or other methods to detect contours of subjects. After this step, researchers were used to apply feature selection algorithms to select more relevant features [34, 35] in order to remove redundancy and noise within features. The usage of kernel functions [36] and Principal Component Analysis (PCA) [37] was an alternative common scenario. Kernel functions map features in a higher space dimension trying to make them linearly separable, while PCA reduces the size of features and finds correlation, lowering the time needed to train classifiers. Due to the handcrafted feature extraction phase, this pipeline is very inefficient and error-prone: often, obtained performances were very low not because of the selected classifier being not able to classify patterns, but because features extracted were not representative for the domain of the problem or were not the most discriminating ones.

For text classification, the approach was very similar to the one explained for images: the main difference is in the feature extraction step, where different methods are employed. Dealing with text, a very common feature extraction algorithm is of Bag-of-Words (BoW). It creates a vector representation for a document, first creating a dictionary of all words of documents and then assigning a “1” value if a word occurs in the document else “0”. Another common technique is term frequency (TF) which assigns a weight to each word counting the number of its occurrences. TF is often used in conjunction with inverse document frequency (IDF) which computes the score of relevancy for each word. This two measures are merged together, computing the product by TF and IDF. Intuitively, if a word occurs many times in a document, it is considered relevant for that document, however, if it occurs in almost all documents, it means is not so relevant for understanding the topic of a text. The disadvantage of all these approaches is the casting of a document into a dictionary vector where correlations between words are lost and no information about semantics is kept.

To solve this problem Word2Vec (W2V) [38] has been introduced. Starting from a corpus of documents, it is able to create a real-valued vector for each word in documents. Its strength is the capability of capturing semantic relationships between words: words occurring in similar contexts will have similar vector representations. On top of w2v is built Doc2Vec (D2V): it has the same objective of Word2Vec, but instead of dealing with single words, it deals with entire documents. Thus each document in a corpus is represented with a fixed length vector keeping the same property of Word2Vec.

2.2 Convolutional Neural Networks

In recent times, deep neural networks have lead to outstanding results on a variety of pattern recognition tasks, such as computer vision and voice recognition [31]. One of the essential components leading to these results is a special kind of neural network called CNN. CNNs are inspired from visual cortex. It has small regions of cells sensitive to specific regions of the visual field, for example to edges or orientations. An experiment performed in 1962 by Hubel and Wiesel [39] showed that some individual neuronal cells in the brain respond only when exposed to vertical edges, while others respond only when seeing horizontal or diagonal edges. Hubel and Wiesel discovered that these neurons are organized in a hierarchical way and, together, they can produce a visual perception. This is the base concept behind each kind of CNN. In Deep Learning these regions are called receptive fields. Receptive Fields are represented as weighted matrices, also known as kernels, that are sensitive to similar local subregions of an image. The degree of similarity between a region of an image and a kernel can be computed convolving the kernel on the region.
For example, a 3×3 kernel like:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{2.1}$$

can be convolved over one image to detect the presence of edges as shown in Figure 2.1. In a traditional feed-forward neural network each input neuron is connected to each output neuron of the next layer. This kind of network is also called fully connected network. In CNNs, convolutional layers applied to the input image are used to compute the output. This leads to obtain local connections, where each region in the input is bound to a neuron in the output. Typically in a CNN there are many layers that apply different filters, as the one shown in 2.1, and combine results.

Pooling layers are usually applied after convolutional layers. The purpose of these layers is to reduce the dimensionality of the input, selecting a value from a local region. The typical operation performed is the max-pooling, thus for a region the maximum value is selected, however, different selection criteria can be used depending upon the scope of the network.

During the training step, a CNN automatically learns the values of its filters based on the task. For example, in image classification, it may learn to detect edges from raw pixel values in the first layer, then it uses edges to detect shapes and finally leverages on features extracted to find higher level features that can be exploited in the last layer to perform the final classification.

This architecture has two important aspects: location invariance and compositionality. The location invariance allows, for example, to classify whether or not there is a specific object in an image. Filters are convolved over the entire image, thus the position of the object to be detected is not important. Rotation and scaling are performed applying pooling filters. Moreover invariance to translation is obtained. The second aspect, local compositionality, is obtained because each filter composes a local patch of lower level features into higher level representation. From basic structures of an image like edges, gradients and blurs, kernels start to learn more highly discriminating complex features for the processed image. In Figure 2.1, taken from [40], some sample images from all layers of a network similar to the famous AlexNet [31], pointing out information learned by kernels.

Basically CNNs are composed of a series of convolutive layers with non-linear activation functions, pooling and fully connected layers. As described, CNNs use convolution operators. Given an image x with k channels and a kernel w, the convolution operator creates a new image y in the following way:

$$y_{i'j'k'} = \sum_{ijk} w_{ijkk'} x_{i+i',j+j',k} \tag{2.2}$$

where k' corresponds to the index of filters/kernels in the convolution.

A linear filter is followed by a non-linear activation function applied identically to each component of a feature map. The most used non-linear activation function is the Rectified Linear Unit (ReLU) and it is defined as follow:

$$y_{ijk} = \max\{0, x_{ijk}\}. \tag{2.3}$$

Another key aspect of Convolutional Neural Networks are pooling layers. They subsample their input through a pooling operators which deals with individual feature channels, merging nearby feature values into one. The most common way to perform pooling is to apply a max function (max-pooling) to the output of each filter. Max pooling is defined as follow:

$$y_{ijk} = \max\{y_{i'j'k} : i \leq i' < i+p, j \leq j' < j+p\}. \tag{2.4}$$

Pooling reduces the output dimensionality but keeps the most salient information and it also provides a fixed-size output matrix, which typically is required for classification. In image recognition, pooling also provides basic translation invariance and rotation properties.

The success of CNNs in computer vision started few years ago with the AlexNet [31]. This network was trained on the ILSVRC-2012 training data, containing 1.2 million training images belonging to 1000 classes. It was able to half the error rate on that task, strongly beating traditional hand-crafted approaches. As shown in Figure 2.2, the net is composed of five convolutional layers, some of them followed by max-pooling layers, and three fully-connected layers with a final softmax. Authors applied dropout to reduce

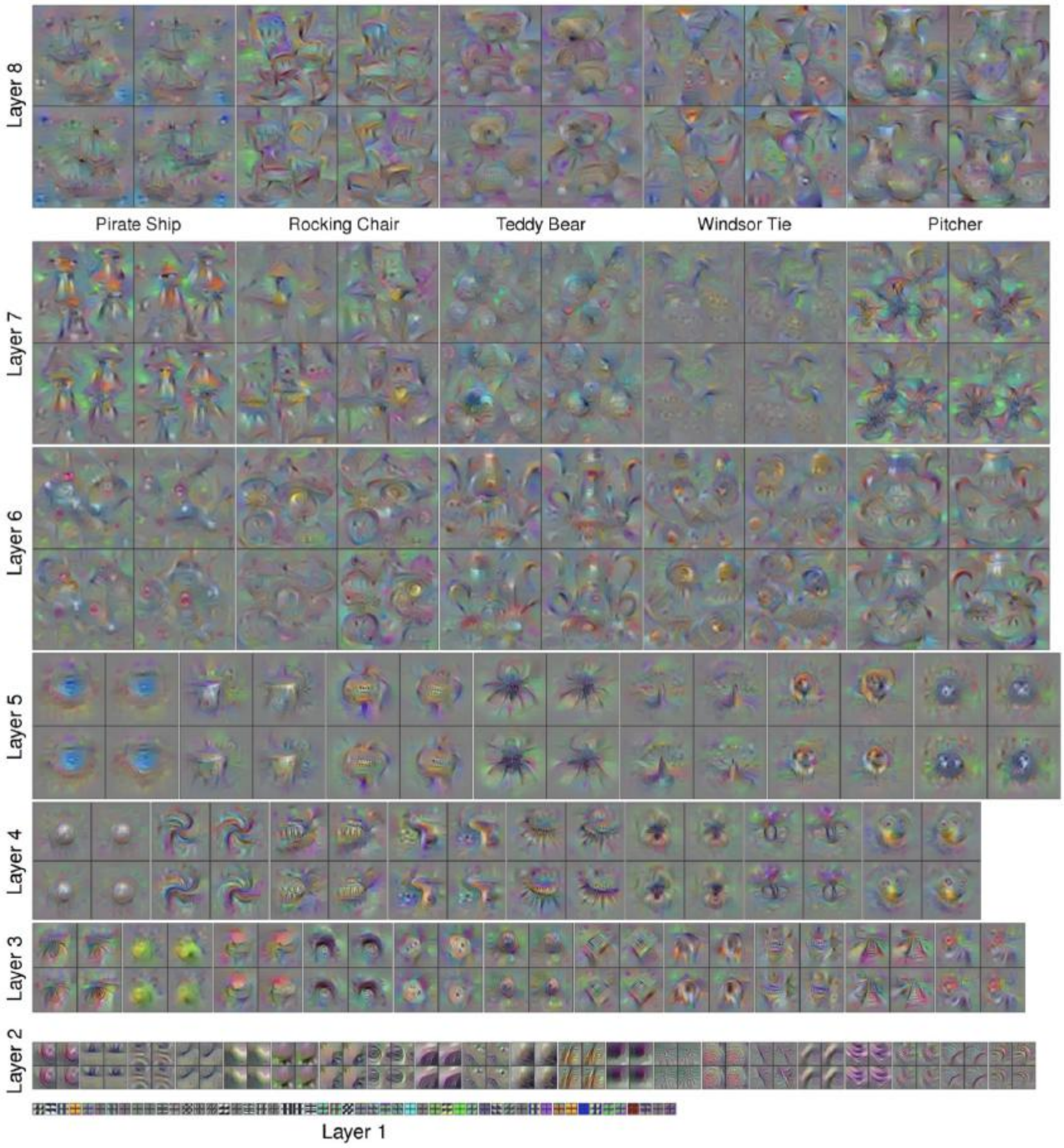


Figure 2.1: Sample features of eight layers of a CNN. For each layer the original size is kept. In lower layers sizes are larger and features seem to be noise, because only edges, blurs and gradients are detected. Going toward the last layer, it can be noticed that sizes are smaller but features resemble to objects of interest from the dataset. Translation invariance and compositionality are more evident in higher layers.

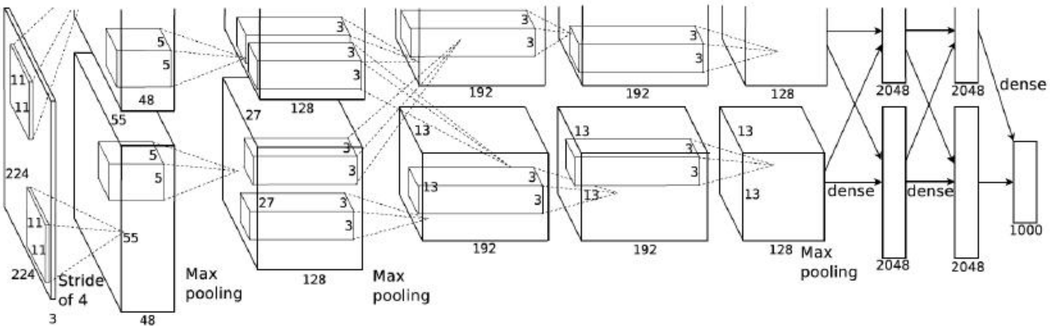


Figure 2.2: AlexNet architecture proposed by Alex Krizhevsky taken from its original paper. AlexNet consists of five convolutional layers followed by three fully-connected layers.

overfitting in the fully-connected layers. Dropout is a powerful regularization method introduced in [41], which has shown benefits for large neural networks. The simple key concept of dropout is to reduce co-adaptation between units. This objective is obtained randomly dropping units and their connections during the training phase.

The performance of AlexNet motivated a number of CNN-based approaches, all aimed at improving performance over this model that became the de-facto standard in computer vision. Just like AlexNet was the winner for ILSVRC challenge in 2012, a novel CNN-based net [42] was the best at ILSVRC-2013. The key insight was the training of a network to jointly classify, locate and detect objects in images. This led to a boost of classification, detection and localization accuracies. GoogleNet [43], won the ILSVRC-2014, establishing the fact that very deep networks can reach higher accuracies in classification task. Authors introduced a trivial 1×1 convolutional layer after a regular convolutional layer, this reduced the number of parameters and resulted in a more expressive power. Further details are reported in the original paper [44], where authors show that having one or more 1×1 convolutional layers is similar to have a multilayer perceptron network processing the outputs of a convolutional layer that precedes it. VGG-19 [45] is another example of highly performing CNN. An interesting feature of VGG architecture is that it replaces larger sizes convolutional filters with a stack of smaller sized filters. These smaller sized filters, typically, are selected in a way that they have approximately the same number of parameters as the larger ones they are replacing. This design decision provided efficiency and regularization-like effect on parameters due to the smaller size of filters involved.

2.3 Convolutional Neural Network for Natural Language Processing

Thanks to stunning results obtained on image classification tasks, Convolutional Neural Networks were also applied to Natural Language Processing (NLP) problems, achieving good results. In NLP tasks, typically, the input is made up of a set of sentences or a corpus of document. Each document must be transformed into a fixed length matrix where each row represents the encoding of a word through a vector of real numbers. The common word2vec [38] approach converts each word in a vector of a specified length, however, it is not the only one available; similarly GloVe [46] or one-hot vectors indexing dictionary can also be used. Convolutional nets applied to NLP, usually perform the so called 1-D convolution operation. Differently from images, where kernels slide from left to right over an entire image, in NLP tasks, kernels have the same length of the encoding method selected, while the height, or region size, may vary. This results in applying the convolution kernel in a single direction only: from top to bottom of a text document.

CNNs for NLP are trained to address classification tasks, such as Topic Categorization, Spam Detection and Sentiment Analysis. Yoon Kim [47] introduced a CNN architecture for Sentiment Analysis and Topic Categorization tasks. It is composed of an input layer which takes a sentence in a form of concatenated word2vec word embeddings, followed by a convolutional layer with multiple filters, then a max-pooling layer and finally a softmax classifier. Kim evaluated this model on many datasets achieving good performance and, in some cases, state-of-the-art results.

Johnson and Zhang [48] trained a CNN directly to one-hot vectors from scratch, differently from word2vec or GloVe approaches that need to be trained on documents. Authors also introduced a more compact bag-of-words-like representation to reduce the number of parameter the network needs to learn. This effect occurs because there is a strong correlation between the number of parameters to learn and the cardinality of the dataset: more samples are required to train a model with larger number of parameters. In [49] authors extend the model with an additional "region embedding" learned using a CNN predicting the context of text regions. Approaches proposed in these papers seem to work well for long texts, however, performance on short text are not clear. To deal with short texts, like tweets, it seems to be a good idea to exploit the capability of a pre-trained word embedding to overcome the lack of information from the text.

Using CNNs in conjunctions with other approaches designed for text like word2vec, GloVe or one-hot vectors, require the choice of several hyperparameters such as embedding length (word2vec, GloVe, one-hot), sizes of convolutional kernels and pooling layers with their own strategies (max, average) and activation functions (tanh, ReLU). The work in [50] evaluates empirically the effect of above-mentioned hyperparameters on CNNs. Results show that max-pooling performs better than average pooling, while filter sizes are important but task-dependent and that regularizazion does not have huge impact on NLP tasks.

The main work of this thesis, explained in Section 3.2, is inspired from the architecture, introduced for NLP, shown in Figure 2.3 [50]. The original model has been slightly changed. The embedding computed by the model is 128-dimensional. Moreover we use it as feature extractor for the text: in the fully-connected layer we extract the text feature that will be represented onto images. All these transformation are applied on the text, while images are resized to the desired output size. Once we preprocessed all images, applying the information coming from the text in the upper part, dataset is ready to be trained with a "standard" CNN, designed for image classification.

2.4 Datasets

In this Section, datasets employed in the thesis are presented. In the Subsection 2.4.1 datasets used to evaluate the multimodal document classification are shown, while in Subsection 2.4.2 datasets of experiments on Git loss function are explained.

2.4.1 Datasets for Multimodal Document Classification

We tested our approach on two multimodal datasets coming from different topics.

The first dataset, UMPC Food-101, consists of images and text descriptions of food recipes from all over the world, as the name suggests. Text description are written in English. It contains about 100000 items of food recipes belonging to 101 classes. This dataset is collected from the web and each item consists of an image and the HTML webpage on which it was found. We extracted the title from HTML document tags and use it as text description. In this dataset there are 101 classes of recipes taken from the most popular categories of the food picture sharing website [1].

The second dataset, called *Ferramenta* dataset, has been provided by an Italian online price comparison company. Through their platform, this company provides a service which let customers to quickly compare the best prices for different products, ranging from mobile phones to fridge and from gardening tools to hardware tools. We collected a set of data from the hardware tools category (Ferramenta) and thanks to a set of 3 experts, all commercial advertises from this category have been tagged. A total of 88010 offers, composed of images and textual descriptions, has then been split into 66141 and 21869 for train and test sets respectively.

[1] www.foodspotting.com

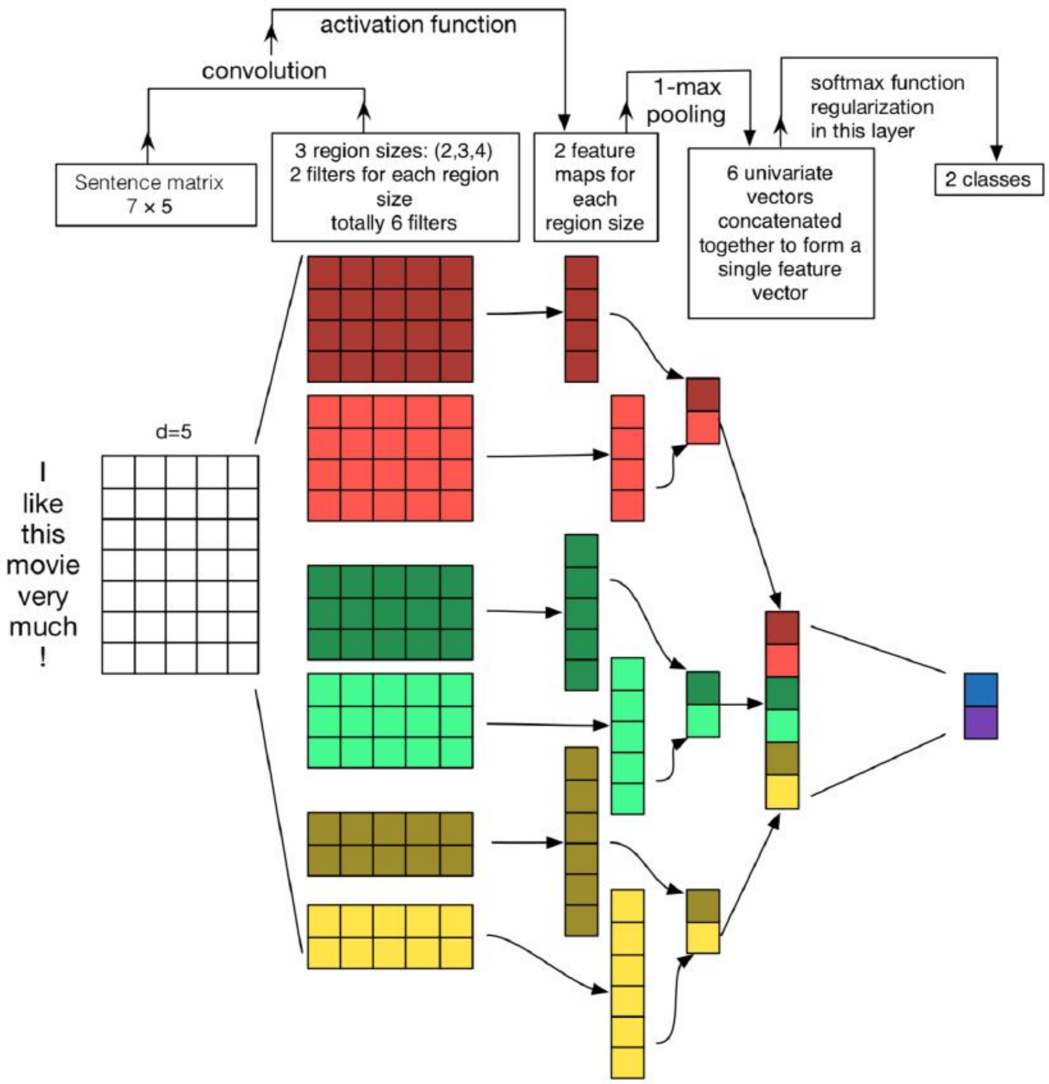

activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

Figure 2.3: Illustration of a CNN architecture for sentence classification. We depict three filter region sizes: 2, 3 and 4, each of which has 2 filters. Filters perform convolutions on the sentence matrix and generate (variable-length) feature maps; 1-max pooling is performed over each map, i.e., the largest number from each feature map is recorded. Thus a univariate feature vector is generated from all six maps, and these 6 features are concatenated to form a feature vector for the penultimate layer. The final softmax layer then receives this feature vector as input and uses it to classify the sentence; here we assume binary classification and hence depict two possible output states. Source: Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification.

2.4.2 Datasets for Discriminative Feature Learning

We evaluated the Git loss function on two famous face recognition benchmark datasets: LFW [51] and YTF [52] in unconstrained environments i.e. under open-set protocol. LFW dataset contains 13233 web-collected images from 5749 different identities, with large variations in pose, expression and illumination. We follow the standard protocol of *unrestricted with labeled outside data* and tested on 6000 face pairs. Results are shown in Table 4.3.

YTF dataset consists of 3425 videos of 1595 different people, with an average of 2.15 videos per person. The duration of each video varies from 48 to 6070 frames, with an average length of 181.3 frames. We follow the same protocol and reported results on 5000 video pairs.

3 Preliminary Works

In this Chapter, I report salient preliminary works that acted as foundation for the main proposals of this thesis. One of the main studies is related to multimodal classification exploiting images and text, thus, it was necessary to study both modalities before merging the acquired knowledge within a single method. The other main contribution is a novel loss function which minimizes intra-class variation and improves inter-class distance. Separability and discrimination is a point to keep in mind in classification tasks to achieve good performance. Even though, it has been developed apparently from a unrelated topic, i.e. face recognition, the proposed approach is easily adaptable to any different task without any modifications. Thus it can be also employed with the multimodal approach proposed in Chapter 4.

All approaches presented have been developed in collaboration with my colleagues of ArteLab[1]. So, during the explanation of the approaches, I will use the first plural person to reflect this collaboration.

In this section, three preliminary works will be summarized [53, 54, 55].

The first one is an approach introduced to classify multimodal documents containing images and text using traditional classifiers. The aim was to understand how simple single-modal classifiers perform when combined together for multimodal classification tasks. We, then, compared results with multimodal approaches.

In the second work, a simple but very effective pre-process step that we introduced to perform text classification task with CNN, typically used for image classification, is

[1] http://artelab.dista.uninsubria.it

shown. This represented one fundamental aspect for the main study of this thesis.

After having developed a base knowledge on text classification field, we studied in deep image classification to have a more complete view of the literature and state-of-the-art models, which is useful for better understanding the multimodal classification topic. We tackled the object recognition and detection fields, studying existing models and adapting them to the context of our interest. At the very end of this Chapter, I will discuss about a pipeline we introduced for object detection and object recognition.

Next sections are organized as follow: in Section 3.1 a summary of our work named "Multimodal Classification in Real-World Scenarios" [53], while Section 3.2 contains an overview of our paper "Semantic Text Encoding for Text Classification using Convolutional Neural Networks" [54]. Both papers have been presented at the International Conference on Document Analysis and Recognition (ICDAR-WML 2017). Section 3.3 illustrates a work named "Reading Meter Numbers in the Wild" [55] accepted at Digital Image Computing: Techniques and Applications (DICTA2019).

3.1 Multimodal Classification: a comparative study

In this paper we introduced an approach leveraging on weighting and meta-learning combination methods that integrate the output probabilities obtained from text and visual classifiers. Typically, text or images are used in single way classification, however, many times ambiguities present in either texts or images may reduce the performance. This lead us to combine text and image of an object or a concept in a multimodal approach to enhance the performance of the final classification. We trained text classifiers on Bag-of-Words and Doc2Vec features, while for the visual branch, we extracted vectors from the last fully connected layer of a Deep Convolutional Neural Network. Single-modalities approaches are compared with early fusion and late fusion multimodal methods. Results show that multimodal classification achieves better performance than single-modal classification, especially on very noisy datasets.

3.1.1 Introduction

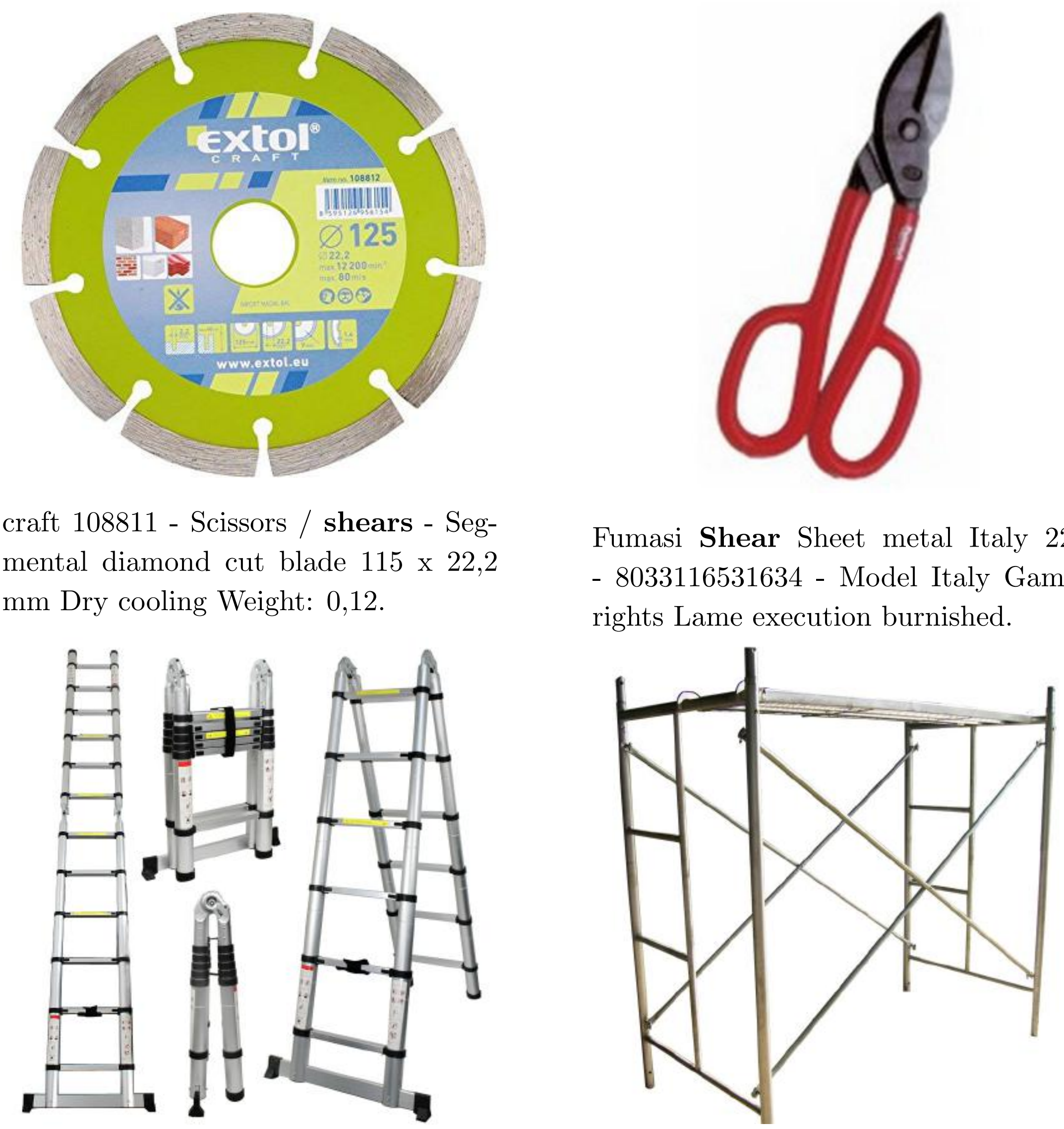
With the rise of e-commerce websites, users are provided information often coming from different sources, for example text and image. For each item on sale, a user can select a product based on a text and an image that show characteristics, colors and other features of the product. However, sometimes, the image and the text of an advertisement are not consistent, which confuses the users that are interested in buying that product. We use different kind of data to perform a multimodal classification, a technique that leverages on features extracted from different modalities to enhance the classification performance. The proposed approach is summarized in Figure 3.3 and uses Convolutional Neural Network (CNN) [31] and other classifiers to achieve the above mentioned goal. This method can obtain high classification accuracy, especially on data characterized by noisy text (grammatically ill formed sentences, short text document, technical details, etc.). Experiments are conducted on advertisements, as shown in Figure 3.1, where the description contains a noisy text and an ambiguous image in some cases.

The image and the text of a document usually contain information describing the same object or concept. In ambiguous situations it is useful to extract the information content from the text and image. For example, in Figure 3.1, the image and the text in the second column describe a pair of shears and a ladder respectively, without ambiguities. But looking at the first column of the same image and we want to classify it only using the text, a classifier may incorrectly classify it as "*shears*". Conversely, if we look at the example in the second row and we want to classify that advertisement only analyzing the image, a classifier may incorrectly annotate it as a "*ladder*". In this way, by combining text and image it is possible to disambiguate wrong classifications and improve the classification result. This leads to the use of a multimodal approach using textual and visual features on a variety of tasks including modeling semantic relatedness, compositionality and classification [56, 57, 58, 59, 60].

The fusion of different modalities generally occurs at two levels: at the level of *features* or *early fusion* and at the level of *decisions* or *late fusion* as described in [61]. Some examples of early fusion such as [62, 60], directly concatenate text and image features to produce a single multimodal vector (see graphical representation in Figure 3.2), obtaining promising performance in other contexts other than classification. Thus, we can show that in certain contexts an early fusion approach results in a classification performance that is better than text or image classifiers: however, it never outperforms better classifiers. In this work, we present two late fusion [61] mechanisms, weighting and meta combination, to combine the output of visual and textual classifiers.

The performance of the two strategies outperforms an early fusion [61] classifier trained on text and visual features concatenation. Visual features are obtained by using CNN extracted features whereas text features are obtained using Doc2Vec (D2V) from [63] and BoW [61]. We also created a dataset called "*Ferramenta*", which contains ambiguous images and noisy text descriptions of commercial offers. Existing datasets, such as [64] are mainly characterized by a couple of labels or keywords associated to an image representing a concept. Our dataset provides images and descriptions representing adverts, that are usually available on an e-commerce website. For the purposes of academic research, we will publish our dataset and we believe that it can be used for a variety of useful tasks. Table 3.2 shows some of the examples with our fusion method on Ferramenta dataset. Examples (a) - (c) show that the method correctly classifies an advert even if one of the models or both make wrong classifications.

In literature it is uncommon to find a dataset with both text and image such as the one presented in this work, which is created through the combined use of text and image



craft 108811 - Scissors / **shears** - Segmental diamond cut blade 115 x 22,2 mm Dry cooling Weight: 0,12.

Fumasi **Shear** Sheet metal Italy 220 - 8033116531634 - Model Italy Gambi rights Lame execution burnished.

Finether 3.2M Portable Aluminum Telescoping **Ladder** with Finger Protection Spacers for Home Loft Office, EN131 Certified, 330 Lb Capacity.

Custom multifunction dynamic construction **scaffolding** (11'6 x 4' x 2'6 Base), simple for decoration -up 150 kg -weights only 16 kg.

Figure 3.1: In the top row, two examples of ambiguous textual descriptions that can be disambiguated through the analysis of the respective images. In the bottom row, two examples of ambiguous images that can be disambiguated through the analysis of the respective description.

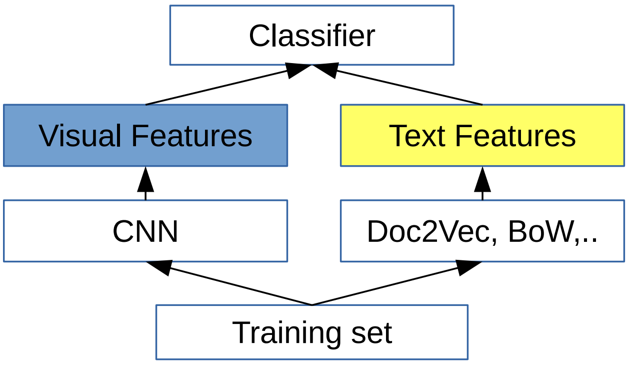


Figure 3.2: A classical early fusion [61] multimodal approach where texts and images are used independently to extract the features in supervised manner or by special operators. These two types of features are concatenated together in order to train a single output classifier.

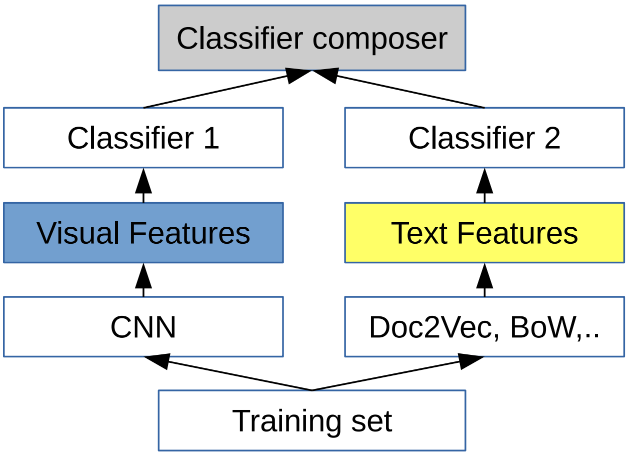


Figure 3.3: The late fusion model. Text and images are used independently to extract the features in a supervised manner or by specific operators. In each of these two types of features, a classifier is trained to output class probabilities. The latter are fused together by special algorithms.

in an advertisement. The majority of the datasets available in literature are related to datasets of images that are then associated to labels to force multimodality, such as PASCAL VOC2007 [64] extended with Flickr tags[2].

3.1.2 Proposed Method

The purpose of supervised learning is to categorize patterns into a set of classes. The main idea behind the ensemble methodology is to weigh several individual classifiers and combine them to obtain a classifier that outperforms individual classifiers, also called late fusion [61] in a multimodal approach. Empirically, ensembles tend to yield better results when there is a significant diversity among models [7]. Many ensemble methods, therefore, seek to promote diversity among the models they combine.

In the ensemble fusion model, texts and images are first processed separately to provide decision-level results, as described in [65, 61]. Results are then combined using two different approaches: weighting methods and meta-learning methods [7]. Weighting methods are useful if the base-classifiers perform the same task and have comparable success. Meta-learning methods are best suited for cases in which certain classifiers consistently correctly classify, or consistently mis-classify, certain instances.

In our model, as in Figure 3.3, text processing and image processing are carried out on text and images separately and a fusion algorithm is used to combine the results. The details of the model and different ensemble approaches are explained below.

Weighting methods

When combining classifiers with weights, a classifier's classification has a strength that is proportional to an assigned weight. This weight can be fixed or dynamically determined for the specific instance to be classified. Suppose we are using probabilistic classifiers, where $P(y = c|x)$ denotes the probability of class c given an instance x. The idea of the *Distribution Summation* (DS) combining method [7] is to sum up the conditional probability vector obtained from each classifier. The selected class is chosen according to the highest value in the total vector. Formally, it can be written

$$class(x) = \operatorname*{arg\,max}_{c_i \in dom(y)} (P_t + P_v) \tag{3.1}$$

where P_t and P_v are the probabilities $P_t(y = c_i|x)$ and $P_v(y = c_i|x)$ of the text classifier and visual classifier respectively.

The weights α_t and α_v of each classifier can be set proportional to its accuracy performance on the training set or validation set, obtaining the following *Performance*

[2] http://lear.inrialpes.fr/people/guillaumin/data.php

Weighting (PW) formula

$$class(x) = \underset{c_i \in dom(y)}{\arg\max}(\alpha_t P_t + \alpha_v P_v) \tag{3.2}$$

where each α_k denotes the weight of the classifier, such that $\alpha_k \geq 0$ and $\sum \alpha_k = 1$.

According to the *Logarithmic Opinion Pool* (LOP) defined in [7], the selection of the preferred class can be also performed in this way:

$$class(x) = \underset{c_i \in dom(y)}{\arg\max}\, e^{\alpha_t log P_t + \alpha_v log P_v} \tag{3.3}$$

Meta-combination methods

Meta-learning means learning from the classifiers produced by the inducers and from the classifications of these classifiers on training data.

In this work we tested the *Stacking* (S) meta-combination method. Stacking is a technique for achieving the highest generalization accuracy [7]. First, two algorithms are trained on images and text descriptions using the available data, then a combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms as additional inputs. If an arbitrary combiner algorithm is used, then stacking can theoretically represent any of the ensemble techniques described above. Stacking performance can be improved by using output probabilities for each class label from the base-level classifiers. Each training instance i of a stacking meta-combiner, consists of a first set of n probabilities $P_{1,t}(y = c_1|x) \dots P_{n,t}(y = c_n|x)$, computed by the model used for the text classification, and concatenated to a second set of n probabilities from the visual model $P_{1,v}(y = c_1|x) \dots P_{n,v}(y = c_n|x)$. All these input probabilities are associated to the same set of binary outcome variables $y_1 \dots y_n$.

We experimented with two Stacking combiner algorithms: a simple Logistic regression (S-L) model with a ridge estimator and a Multilayer Perceptron Classifier (S-MLP) that uses back-propagation to classify instances [7]. These two algorithms are available in the Weka open-source library [66].

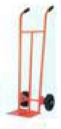
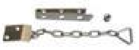
3.1.3 Conclusion

Results obtained indicate that the proposed multimodal setting outperforms classifiers based on an early fusion approaches. In addition to the experiments on Ferramenta dataset, we computed experiments also on PASCAL VOC 2007. All details on the experimental results on this dataset can be found in the original paper [53]. On the basis of the results obtained on Ferramenta and PASCAL VOC2007 dataset, our multimodal setting is recommended for applications where ambiguous text can exploit image to resolve ambiguities and, vice versa, to enhance performance.

Table 3.1: Multimodal fusion accuracy results with the Ferramenta and Pascal VOC2007 test set. In the top half of the table the results of the proposed late fusion model with the best classifiers for each feature. The bottom half shows the results of the best visual model combined with the best textual model, based on the D2V features. The two rows *FConc* (Feature Concatenation) show the results of the early fusion model shown in Figure 3.2, whereas the other rows labeled as *Fusion* show the results of the model shown in Figure 3.3. *Prob* means that the final combiner receives in input the probabilities of the two underlying models.

	features	**algorithm**	**P**(%)	**R**(%)	**F1**(%)	**Acc**(%)
Text	bow1000	RF	91.80	91.70	91.70	91.73
Visual	CNN4096	SVM	90.60	90.30	90.30	90.31
FConc.	CNN+BoW	RF	89.90	88.10	88.00	88.14
Fusion	prob	DS	93.80	93.70	93.70	93.74
Fusion	prob	PW	93.80	93.70	93.70	93.74
Fusion	prob	LOP	**94.60**	**94.40**	**94.40**	**94.42**
Fusion	prob	S-L	90.40	89.50	89.50	89.53
Fusion	prob	S-MLP	93.40	93.20	93.20	93.21
Text	D2V	SVM	88.20	87.40	87.10	87.38
Visual	CNN4096	SVM	90.60	90.30	90.30	90.31
FConc.	CNN+D2V	SVM	90.40	89.50	89.50	89.53
Fusion	prob	DS	92.50	92.40	92.30	92.37
Fusion	prob	PW	92.50	92.40	92.30	92.38
Fusion	prob	LOP	**93.20**	**92.90**	**92.90**	**92.94**
Fusion	prob	S-L	91.80	91.60	91.40	91.57
Fusion	prob	S-MLP	92.00	92.00	91.90	92.03

Table 3.2: From (a) to (f), six classification examples of instances (Description and Image) belonging to the test set of the Ferramenta dataset. The Visual and Text columns represent respectively the output produced by the classifier 1 and 2 showed in Figure 3.3. The output of the final composer is shown in the column Fusion.

(a)	Text	Visual	Fusion	
actual	screwdriver	screwdriver	screwdriver	
predicted	screwdriver	**glue**	screwdriver	
Description	silverline 918547 set 6 screwdrivers, 6 pcs			
(b)	Text	Visual	Fusion	
actual	cart	cart	cart	
predicted	**scissor**	cart	cart	
Description	cart box trolleys with solid tires			
(c)	Text	Visual	Fusion	
actual	screwdriver	screwdriver	screwdriver	
predicted	**socket wrench**	**cart**	screwdriver	
Description	ks tools 159.1203 screwdriver ergotorque plus key 5.5 mm., ks tools 159.1203 screwdriver key ergotorqueplus 5.5 mm.			
(d)	Text	Visual	Fusion	
actual	safe	safe	safe	
predicted	safe	safe	safe	
Description	88352 staco safe measure s, 88352 staco safe measure s			
(e)	Text	Visual	Fusion	
actual	chain	chain	chain	
predicted	chain	**circular saw blade**	**circular saw blade**	
Description	yale p1040sc deadbolt door locks high security chrome trim, yale locks p1040sc deadbolt door high security chrome trim			
(f)	Text	Visual	Fusion	
actual	nail	nail	nail	
predicted	**screw**	nail	**screw**	
Description	Hardware bulk pack of 20 nails for masonry 3 x 70 mm, bulk pack of 20 Hardware nails for masonry 3 x 70 mm			

3.2 Semantic Text Encoding for Text Classification using Convolutional Neural Networks

We encoded the semantics of a text document into an image to take advantage of the Convolutional Neural Networks (CNNs) architectures that are successfully employed in image classification. We used Word2Vec, which is an estimation of word representation in a vector space that can maintain the semantic and syntactic relationships among words. Word2Vec vectors are then transformed into graphical words representing sequence of words in the text document. Next, encoded images are classified using the AlexNet architecture [31]. Results obtained indicate that the proposed scheme achieves better performance than traditional classifiers built on top of Doc2Vec features.

3.2.1 Introduction

Semantics plays a crucial role to understand the meaning of a text document: humans can understand the semantics easily, however, for a computer, understanding of the semantics is not a trivial task. State-of-the-art word embedding models such as Word2Vec [38] are a step forward to understand the semantics of the text. Word2Vec takes words or phrases from the vocabulary and maps them into vectors of real numbers. Thanks to its objective function, it causes words occurring in similar contexts to have similar word embeddings. We exploit this observation in our method, as shown in Fig. 3.4, where similar word embeddings can be transformed into visual domain with similar encoding which then can be used in text classification with CNNs. We introduced a new encoding scheme called *Semantic Text Encoding* or *ste2img* which encodes Word2Vec features of a text document into an image, capitalizing the abilities of CNNs for classification. Our proposed encoding scheme captures typical computer vision location invariant property along with compositionality. In fact, a CNN intuitively can understand words from pixels, sentences from words, and more complex concepts from sentences.

In literature, many approaches have been introduced to perform text classification exploiting capabilities of state-of-the-art CNN. Zhang *et al.* [22], created a CNN model working with texts at the level of characters. They showed that their approach achieved low error rates compared to traditional approaches. However, in our work, instead of using a convolutional model developed specifically to deal with text classification, we designed a preprocess step which allows us to use the CNNs typically used for image classification.

Kalchbrenner *et al.* [67] created a new model of CNN that used a specialized operation named Dynamic k-Max Pooling to classify sentences of varying length. Moreover, they provided a way to keep word order information intact unlike Bag-of-Words. In addition, they concluded their approach suffered from complexity issues with larger dictionaries,

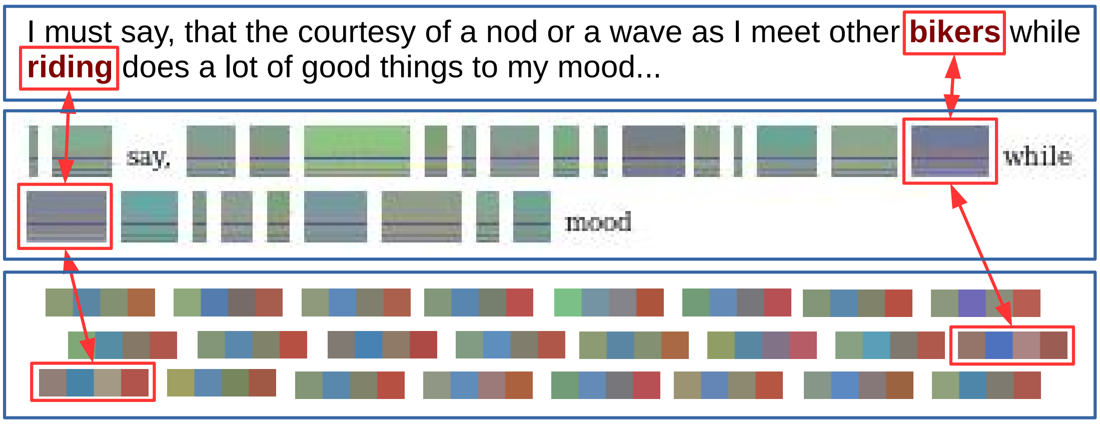

I must say, that the courtesy of a nod or a wave as I meet other bikers while riding does a lot of good things to my mood...
say,
while
mood

Figure 3.4: When analyzing the text, a human can understand that the two words ”bikers” and ”riding” are semantically related and that the document talks about motorcycles. Combining the ability of Word2Vec to find semantic relationships between words and the ability of CNN in images classification, a computer can almost imitate those human capabilities. The visual words encoded by Word2Vec and displayed at the center and on the bottom of this figure, can be easily understood by a CNN. In this encoding, a 12 dimensions Word2Vec feature vector has been transformed into a sequence of 4 RGB colors.

resulting in the reduction of either the dimension of the vocabulary or the length of feature vectors. This leads us to consider that we should have the same issue; in fact, some of the datasets that we used are characterized by a large number of words, which forced us to limit the number of words, especially for the 20news dataset.

Tang *et al.* [68] faced up the sentiment classification problem on Twitter datasets using an ad-hoc model made up of three neural networks to encode sentiment information from text into loss function. To achieve this goal, this study exploits the capabilities of Word2Vec model. They concluded that Word2Vec is not suitable for sentiment analysis tasks as it cannot differentiate the meaning between adjectives before a noun, for example “good” and “bad”. On the other hand, we are interested in the relationships between words rather than polarity, and so this Word2Vec embedding is perfect for our purpose.

3.2.2 Proposed Method

The encoding method described in this section is used to analyze the behavior of a CNN, with various encoding parameters.

We exploit Word2Vec [38] word embedding to get the semantics associated with a text document in an image and then the encoded image can be used in the classification process. As shown in Fig. 3.6, we used a dictionary $F(t_k, v_k)$ where each word t_k, used

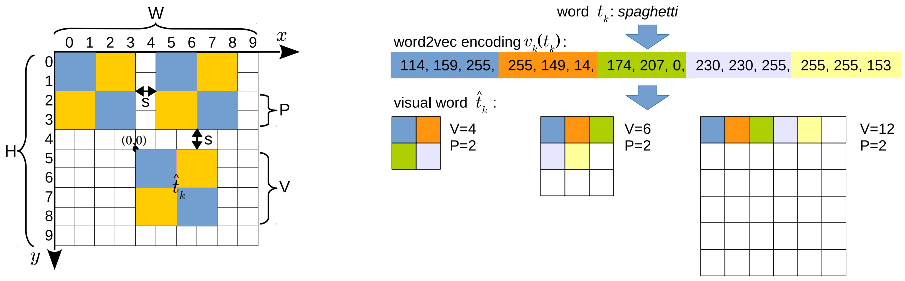


Figure 3.5: A schematic example of our encoding scheme. The figure on the left shows a simple encoding image of size $W \times H = 9 \times 9$, with three visual words $\hat{t}_k$ of size $V \times V = 4 \times 4$ which may contains a maximum number of $(V/P)^2 = (4/2)^2 = 4$ superpixels of size $P \times P = 2 \times 2$. On the right, for example the word "*spaghetti*" is encoded in a sequence of 15 numbers using word2vec, and the same sequence can be transformed into different visual words by changing parameter V (in this example $V = 4, 6, 12$).

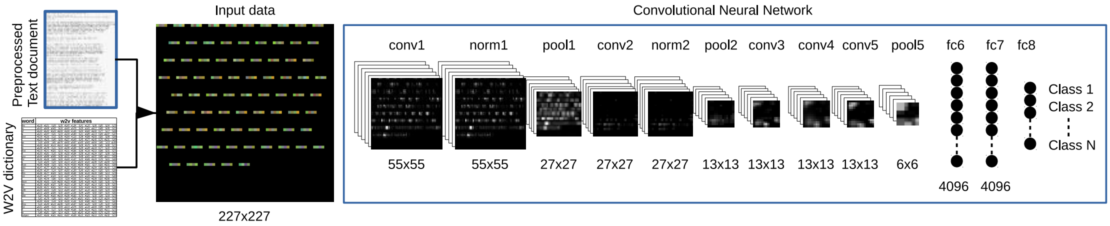


Figure 3.6: Image of the proposed pipeline. Starting from a text, we encode each word with values extracted from a dictionary of feature vectors (W2V dictionary). Values obtained are interpreted as RGB values and represented in the final image. After this preprocess step, the image is classified by a properly trained CNN implementing the AlexNet architecture.

to train Word2Vec, is associated with a vector $v_k(t_k)$ of features obtained from a trained version of Word2Vec.

As shown in Fig. 3.6, to create the representation of a text document D_i, we start from a pre-processed version of D_i, applying some transformations to the text. We applied different pre-processing methods and compared them with the raw text.

To encode all the words $t_k \in D_i$ into an image of size $W \times H$ we introduced a basic scheme that, varying some parameters, allows us to obtain different arrangements of visual words $\hat{t}_k$ in the image.

We introduce the concept of *super-pixel* as a square area of size $P \times P$ pixels with uniform color representing a contiguous sequence of features $(v_{k,j}, v_{k,j+1}, v_{k,j+2})$ extracted as a sub-vector of v_k. Each $v_{k,j}$ component is normalized with respect to all k, in such a way that it can assume values in the interval $[0 \ldots 255]$. Given a word t_k, a *visual word* $\hat{t}_k$ is a square area of $V \times V$ pixels that can contain a maximum number of super-pixels equal to $(V/P)^2$. The position $(0,0)$ of each $\hat{t}_k$ was fixed in the upper left corner. Consequently, each $\hat{t}_k$ is placed in the following image coordinate (x, y):

$$\begin{aligned} x &= k(V+s) \mod (W-V) \\ y &= (V+s)\tfrac{k(V+s)}{(W-V)} \end{aligned} \tag{3.4}$$

where s is the space (horizontal or vertical) in pixels existing between two close visual words. Fig. 3.5 shows an example of an image encoding a text document (on the left), and one example of encoded visual word (on the right). The visual words positioning described in Equation (3.4), produces a wanted vertical misalignment, avoiding regularity in the encoding image.

As shown in Fig. 3.6, each convolutional layer produces different convolutive maps and each of them, from the closest layer to the input up to the last one, produces activation areas that shows how the model understands the semantics of the text.

The first convolutional layer recognizes some particular features of visual words, while remaining CNN layers, aggregate these simple activations to create increasingly complex relationships between words or parts of a sentence in a document.

3.2.3 Experiments

The aims of the experiments are:

- To validate the proposed encoding scheme and understand different parameters and configurations

- To compare the proposed encoding scheme with text classification based on Doc2Vec and SVM

In our experiments we measured the performance of models using overall classification accuracy. The encoding scheme mentioned in Section 3.2.2 produces encoded images from Word2Vec features. These encoded images can be classified using CNNs, however, we used the AlexNet architecture in our experiments. The architecture contains eight layers with weights; the first five are convolutional and the remaining three are fully-connected. The output of the last fully-connected layer is fed to softmax which produces a distribution over the class labels, as shown in Fig. 3.6.

We use the publicly available Doc2Vec tool to train word embeddings, and all parameters are set as in [38] to train sentence vectors on 20news-bydate and Text-Ferramenta datasets. However, we use a smaller window size (7), for StackOverflow dataset, as it is composed of short question titles. Normally, Doc2Vec and Word2Vec are trained on a large corpus and used in different contexts. However, in our work, we trained these two tools with the same training set for each dataset used for text classification.

All experiments are conducted using $s = 1$, images of 256×256 and we never removed stop words from the text. We used superpixel, described in Section 3.2.2, set to 4×4 and 8×8 and we were interested in understanding if best results can be obtained by using a kernel size that is smaller or larger than superpixel. We can see that best results are obtained using a low number of features: 12 or 24. We believe that this resulting difference is due to the encoding technique. In fact for low number of features, our approach encodes feature vectors with superpixels in a single row, while, with large number of features, multi-line encoding is used. The impact is given by the convolutional layer, which cannot recognize scrolling images from left to right thus encoding in a new line as if it belonged to a previously seen word.

We compared our encoding scheme with text classification based on Doc2Vec and SVM. We obtained feature vectors from Doc2Vec model for each instance starting from each document of a dataset and saved embedding to a file, one for training and one for testing instances. Finally, instances have been classified with SVM available in the WEKA open source library [66] to compare the results against ste2img as shown in table 3.3. These results indicate that our encoding scheme outperformed text classification based on Doc2Vec and SVM for StackOverflow and Text-Ferramenta datasets. However, 20news-bydate dataset does not produce similar results with our encoding scheme for 75 and 100 features, because we limit the maximum number of words for each document.

3.2.4 Conclusion

Our proposed scheme outperformed text classification based on Doc2vec and SVM on three datasets i.e. StackOverflow, Text-Ferramenta and 20news-bydate. Overall accuracy obtained indicates that the proposed scheme can be applied to any kind of sentences such as technical descriptions, common sentences or ill-formed phrases with different lengths. Moreover, results were an interesting starting point for many Natural Lan-

Table 3.3: Comparison of the accuracy of the Doc2Vec using SVM and the best results obtained with our proposed *ste2img*.

num. features:	12	24	75	100
SVM Text-Ferramenta	71.02	80.47	87.83	88.69
ste2img Text-Ferramenta	**89.90**	**91.60**	**91.84**	**89.38**
SVM StackOverflow	44.80	61.95	68.62	70.62
ste2img StackOverflow	**45.90**	**86.88**	**87.45**	**78.42**
SVM 20news-bydate	91.54	93.33	**95.30**	**95.88**
ste2img 20news-bydate	**94.99**	**94.48**	91.22	89.93

guage Processing works based on CNNs, such as a multimodal approach that could use a single CNN to classify both image and text information. All other performed experiments can be found in the original paper [54].

3.3 Reading Meter Numbers in the Wild

The automatic detection and recognition of text in images is an important challenge for visual understanding. In recent years, deep neural networks [69, 70, 71, 72, 73, 74] have replaced traditional optical character recognition (OCR) based methods for text detection and recognition. In this work, we presented a system for detecting and recognizing various utility meter numbers in the wild from pictures. This task is challenging due to the huge variability in the visual appearance of numbers in the wild on account of a large range of fonts, colors, styles, orientations and arrangements. While this specific task reduces the space of characters that need to be recognized, the complexities associated with text recognition in natural images still apply.

3.3.1 Introduction

The automatic detection and recognition of text in images is an important challenge for visual understanding. In addition environmental factors such as lighting, shadows, specularities and occlusions further complicate the automatic multi-digits numbers detection and recognition tasks, as can be seen from Figure 3.7. The proposed system leverages on deep convolutional neural networks for detection and recognition. For the detection phase, we employed a fully convolutional neural network to perform a pixel-wise classification, while the recognition phase uses another deep neural network to predict the length of meter numbers their values. The advantages of our approach are robustness against severe perspective distortions, different lighting conditions, blurred images and it is also scale invariant.

Text detection in natural images has been tackled several times [75, 70, 76, 71]. All these attempts differ for task addressed or models used for detection and recognition. Many works in literature use Histogram of Gradient (HOG) features to perform text detection. Minetto *et al.* [77] proposed T-HOG: a HOG-based texture descriptor that uses a partition of an image to detect a single line of text. This approach, however, suffers from orientation issues. Given that HOG deals with lines, texts in several orientations become a problem in this approach. Boran *et al.* [76] adopted a more traditional approach, with the joint use of HOG features and support vector machine [78] to detect Chinese words in images. There are many works that employ Maximally Stable Extremal Region (MSER) to perform text detection [79, 80]. Although [79] and [80] solve slightly different tasks, both use MSER-based approaches to perform the detection step. Dai *et al.* [81] uses the traditional two-stage object detection strategy that consists in region proposal extraction with region classification. The problem of these approaches is that regions proposed often have very high cardinality, which thus need to be reduced with the introduction of a strategy to remove false positives.

With the advent of Convolutional Neural Networks (CNNs) and their stunning results, we opt to perform the detection using a deep model [82]. In [70] authors developed a pipeline for text detection and recognition from natural scene images using deep model. The purpose of our work is similar, however, we employ to a more specific task: multi-digit meter numbers detection and recognition in the wild. Next, for the recognition, we used the model proposed by Goodfellow *et al.* [69]. Gómez *et al.*[71] address the same task we solve in this work. Authors used an end-to-end CNN to predict numbers in a meter. Although the approach obtained promising results, it suffers from severe perspective distortions. However, in our approach, after the detection phase we apply image transformations to align it in a horizontal position, making it possible to deal inclinations. After this step, we apply the classification model to obtain predicted values. Moreover, in our approach we have one model for detection and other one model for recognition that deals with various meter typologies, while authors in [71] worked only with mechanical gas meters.

3.3.2 Proposed approach

We graphically present the approach in Fig. 3.8. It is split into two phases: detection and recognition.

Detection phase

This phase is carried out with the model proposed in [82]. We show this phase in Fig. 3.8 with name "CNN-1". This model takes the image resized to 224×224 pixels and produces correspondingly-sized output image with inference. The training set contains pairs of

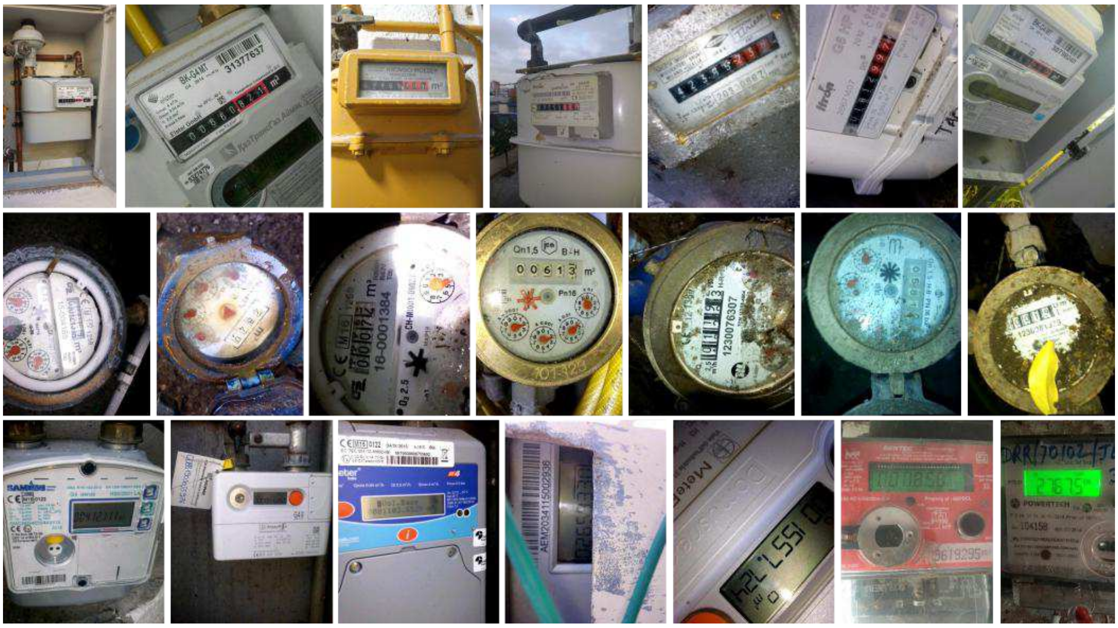


Figure 3.7: Some random images from the dataset. Note that meter typologies are different, ranging from mechanical gas meters to digital electricity meters.

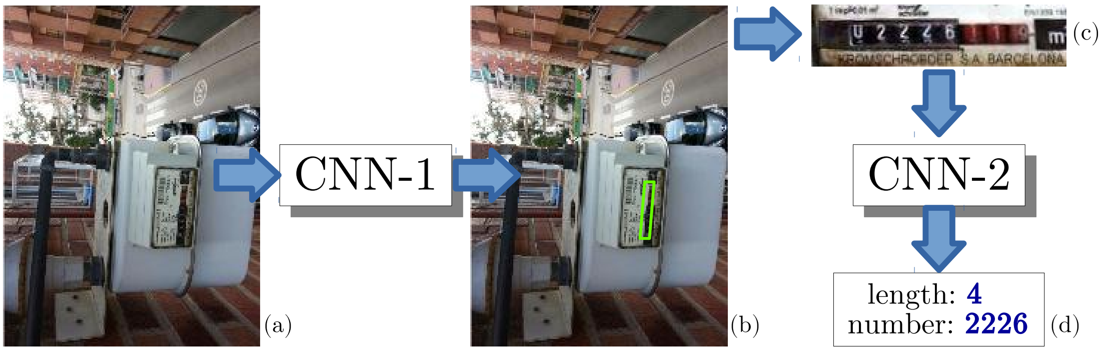


Figure 3.8: Schematic representation of the proposed pipeline. On the left we have the CNN-1 used for detection. It receives as input an image (a) and gives as output an image map with the area of interest (b). On the right the CNN-2 performs the reading starting from the detected region, rotated and cropped (c) producing the reading (d).

images: the original one and the ground truth with pixel values in $\{0, 1\}$, where 0 indicates background and 1 tags numbers. After inference stage, we crop the area of interest from the original image. Furthermore, we rotate the cropped area to obtain a horizontally aligned image. We apply the following strategy starting from the output image obtained from CNN-1 (Fig. 3.8-b):

- extract contours (curves joining continuous points along the boundary with same color or intensity);

- select the best contour that contains the area of interest, keeping the region with the highest ratio $r_i = min(w_i, h_i)/max(w_i, h_i)$, where w_i and h_i are the width and height of the i-th rotated rectangle containing the contour for each of the proposed areas of interest;

- calculate the tilt angle to obtain an horizontally aligned image from the selected area of interest;

- width and height of the selected area are dilated with an increment $d = 0.3 \cdot min(w, h)$ (see the example in Fig. 3.8-c);

- crop the image using the dilated area;

Recognition phase

We perform this step with the model proposed in [69] and name it as "CNN-2" in Fig. 3.8. Note that we preprocess the dataset in the following way:

- for each dilated area, we randomly crop 3 images, containing the full detected region. In this way we obtain a three times larger dataset that improves the overall accuracy of the model.

- resize images to a size of 360×90 pixels. We observed that width is usually 4 times larger than the height.

Once the data is ready, we apply the classifier to get numbers from images. This model is trained with manually assigned labels. For example, if an image contains the value "$00040, 87$", the label would be "40". The model has the ability to discard the decimal and leading zeros parts during training and testing phases. We observed that the majority of numbers in meter images does not exceed more than 5 digits. This led us to set the maximum length to 5.

3.3.3 Experiments

In this section we evaluate performance taking each step individually and finally applying the whole pipeline.

Table 3.4: IoU evaluation with thresholds ranging from 0.1 (Poor) to 0.9 (Excellent) with step size of 0.1.

IoU Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Accuracy	99.37	99.16	98.35	91.06	69.87	40.41	17.42	2.63	0.002

Table 3.5: Accuracy computed on recognition phase for each position: from the 1st to the 5th.

Phase	**1st**	**2nd**	**3rd**	**4th**	**5th**
Recognition	94.60	89.22	92.76	95.10	96.60

Experiment on detection phase

In the first experiment we use images from the detection split. The model detects the area of interest from the input image and we use the mask obtained from the model to calculate Intersection over Union (IoU) metric with the ground truth. IoU is defined in Eq. 3.5:

$$IoU = \frac{AO}{AU} \tag{3.5}$$

where AO is the area of overlapping rectangle between predicted and ground truth, while AU is the area of union rectangle between predicted and ground truth.

Results on detection phase are shown in Table 3.4. Note that, we calculate IoU for various thresholds ranging from 0.1 (Poor) to 0.9 (Excellent). We found out that a value of 0.4 (Good) produces an accuracy of 91.06%, as can be seen from Table 3.4. However, increasing this value leads to lower results. We observed that such situation arises from different ways of annotating the data (see examples available in Fig.3.11).

Experiment on recognition phase

In the second experiment we use images from the recognition split. Note that images used in this experiment contain only the detected area of interest. We compute the overall accuracy considering a prediction right only if all numbers inside the image are classified correctly. In addition, we calculate the accuracy for each number position in the image. The overall accuracy for the recognition phase is 82.70%. We obtain this value from already cropped (6045) test images available in the recognition split. Results are in Table 3.5.

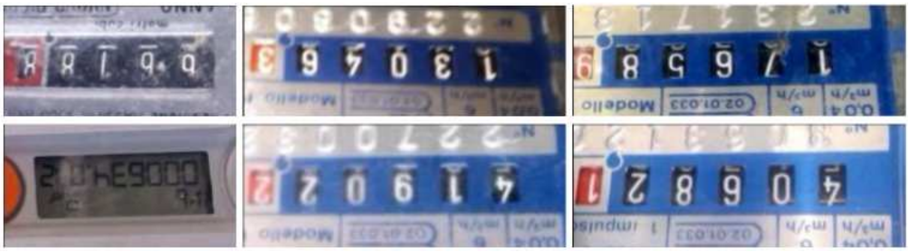


Figure 3.9: Examples of upside down images from the dataset.

Full pipeline experiment

In the third experiment we evaluate the entire pipeline. First, we apply the detection phase to get the area of interest and then we feed it to the recognition phase to obtain the final prediction.

Our pipeline achieves promising results considering that it recognizes numbers from three different meter typologies. Furthermore, the pipeline is capable to produce numbers from mechanical and digital meters. Moreover, our recognition model is more robust than the one proposed in [71], because it can recognize digits from upside down images. In some cases, due to the perspective with which the image is captured, when rotating the area of interest, we obtain a horizontally aligned image but with numbers rotated by 180° degrees as shown in Fig. 3.9.

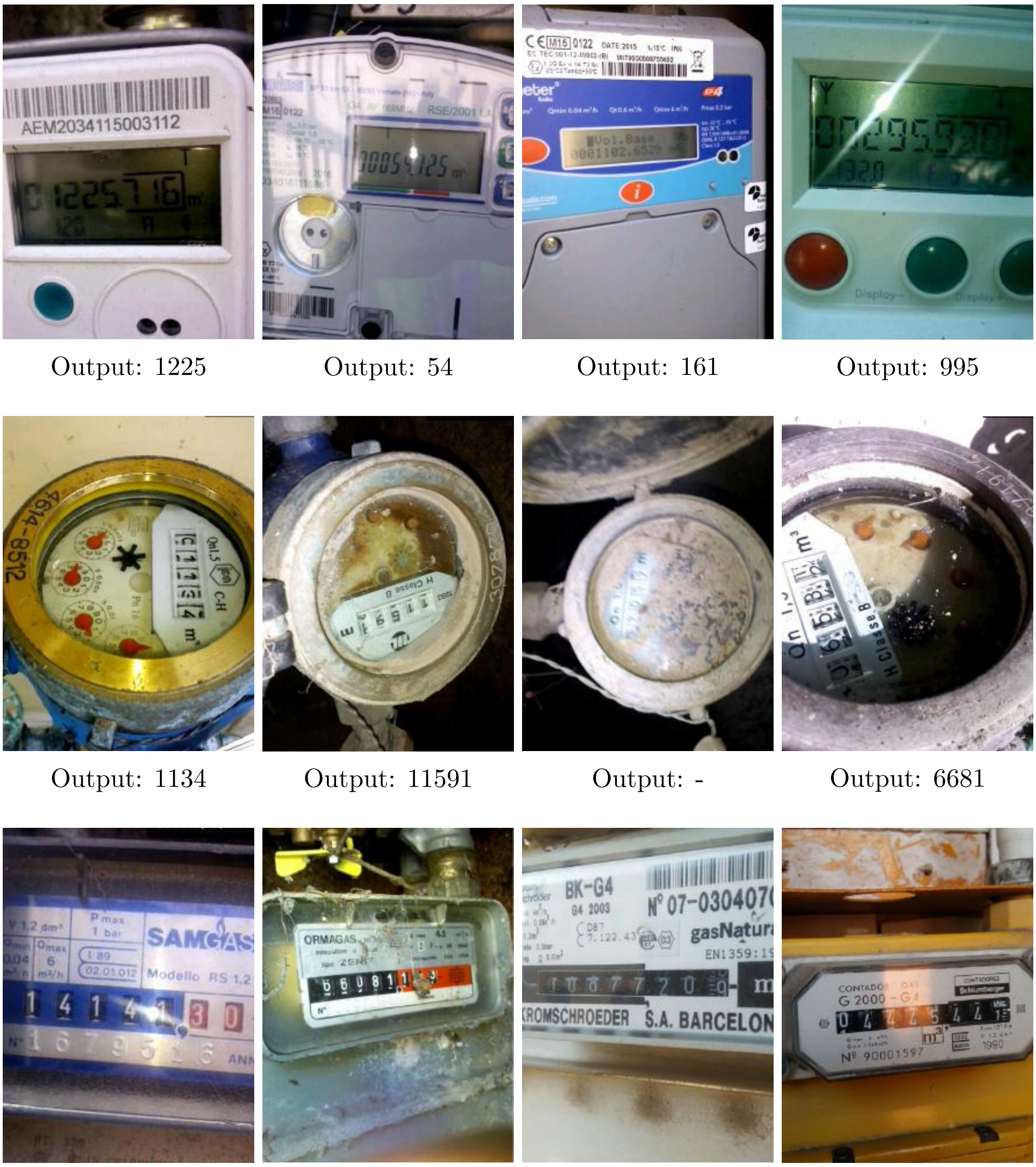
Applying the entire pipeline to the test split, we achieved an accuracy of 85.60%. In addition, we calculate the accuracy of each number based on the position. Results are in Table 3.6. Analyzing results it can be observed that accuracy of the second most significant digit is significantly lower than others. We believe that this happens because of low variability in the dataset for digits in that particular position. We observed that it is difficult to have high variability for most significant digits because most of times they are either “0” or “1”.

Average time required to perform both steps in our pipeline on a single image is of 0.12 seconds on a NVIDIA GeForce GTX 1080 GPU, this means that our approach could also be used in real-time scenarios.

Furthermore, we present qualitative evaluation of the pipeline. Fig. 3.10 shows some examples of correct readings where the performance of the pipeline is particularly robust. In addition, we included some complicated error cases where the pipeline fails to predict the correct output digits.

3.3.4 Conclusion

In this paper we proposed a pipeline to detect and recognize digits of household meters from images. Our method has been tested on different typologies of meter, ranging



Output: 1225 Output: 54 Output: 161 Output: 995

Output: 1134 Output: 11591 Output: - Output: 6681

Output: 14141 Output: 66081 Output: 477 Output: 4745

Figure 3.10: Qualitatively results for various typologies of meter. In each line first two samples are correctly read while remaining show wrongly recognized meters. Strong glares and shadows are the main issues which lead the algorithm to predict wrong results.

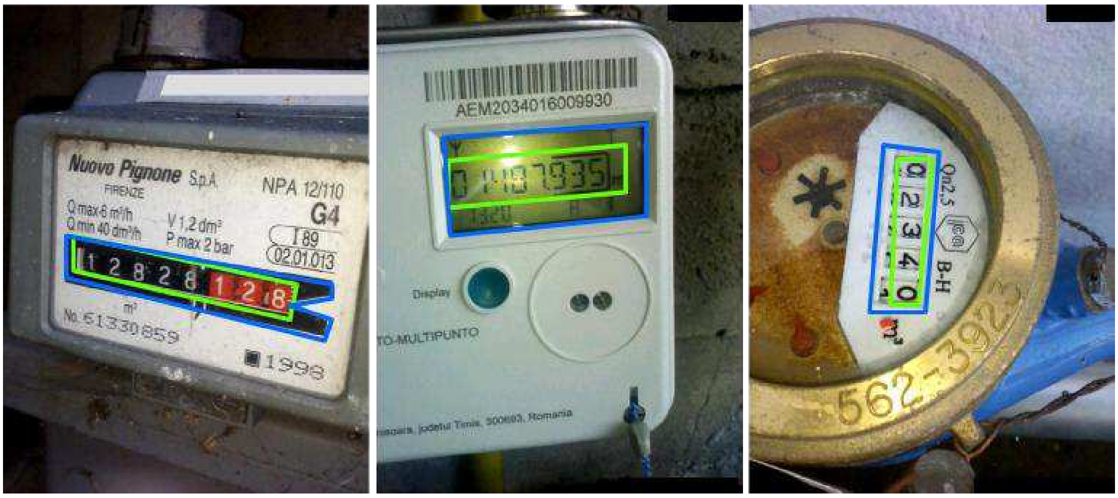


Figure 3.11: Examples of images with superimposed ground truth (area with green borders) and the relative prediction of the CNN-1 model (area with blue borders). These examples highlight the lack of a clear boundary in the area of interest and consequently the dependency of the ground truth from the annotator.

Table 3.6: Accuracy computed on the full pipeline.

Phase	**1st**	**2nd**	**3rd**	**4th**	**5th**	**Acc.**
Pipeline	92.94	88.99	90.95	92.24	92.92	85.60

from mechanical to electric meters of various kind: gas, water and electricity using two deep models, one for detection and one for recognition. The proposed pipeline is robust against severe perspective distortions, different scales and even upside-down images. Results obtained are very promising and the execution time required to apply pipeline makes it possible to use it in real-time applications.

4
Multimodal Document Classification

This chapter contains an overview of the paper ”Image and Encoded Text Fusion for Multimodal Classification” [9] presented at the Digital Image Computing: Techniques and Applications (DICTA 2018).

4.1 Introduction

With the rapid rise of e-commerce, the web has increasingly become multimodal, making the question of multimodal strategy ever more important. However, modalities in multimodal approach come from different input sources (text/image [53, 60, 83] , audio/video [84] etc.) and are often characterized by distinct statistical properties, making it difficult to create a joint representation that uniquely captures the “concept” in the real-world applications. There are many samples where exploiting a single modality to classify a document, it is not possible to get the correct label. In fact, there are document with very similar images but different text description and vice versa. In these cases, it is mandatory to have a method which can deal with multimodal information in order to get the correct label for a document.

This lead us to create a joint representation of an image and text description for this classification problem. Multimodal strategy can exploit such scenario to remove ambiguity and improve classification performance.

The use of multimodal approach based on image and text features is extensively employed on a variety of tasks including modeling semantic relatedness, compositionality, classification and retrieval [56, 60, 59, 57, 83, 85]. Typically, in multimodal approaches, image features are extracted using CNNs. Whereas, to generate text features, Bag-of-Words models or Log-linear Skip-gram Models [63] 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 [86].

With this work, we present a novel strategy which combines a text encoding schema to fuse text features and image in an unified information enriched image. We merge both text encoding and image into a single source so that it can be used with a CNN. We demonstrate that by adding encoded text information in an image, multimodal classification results can be improved compared to the best one obtained on a uni-modality (image/text).

Intuitively superimposing text descriptions onto images may not sound motivating due to several reasons. Since the idea is overlaying the encoded text description onto an image, it might affect the image perception in general. However, the main strength of the approach is that embedded text can be overlaid onto the image with fixed width regardless of the size of text description. We experiment with different embedding sizes to verify that image perception by the network is not affected. Figure 4.4 plots different embedding sizes to explain how the network behaves.

There are two general multimodal fusion strategies to fuse text and images features, namely early fusion and late fusion [61, 86], each one having its advantages and disadvantages.

Early fusion is an initial attempt by the researchers towards multimodal representation learning. The main benefit of early fusion is that it can learn to exploit the correlation and interactions between low level features of each modality. Early fusion methods concatenates text and image features into a single vector which is used as input pattern for the final classifier. The technique is employed for various tasks [62, 60, 83].

In contrast, late fusion [87] uses decision values from each modal and fuses them using a fusion mechanism. Multiple works [88, 89] employ different fusion mechanisms such as averaging, voting schemes, variance etc. The work in [53] showcased a comparative study of early and late fusion multimodal methods. Late fusion produced better performance compared to early fusion method, however, it comes with the price of an increased learning effort. In addition, a strategy must be introduced to assign a weight to each classifier employed. This presents another challenge in late fusion strategy. Our method is inspired from early fusion [60], however, taking advantage of the idea of our previous work [54] 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. To write the textual features on the image, we use an encoding very similar to the one proposed in [54]. The main difference lies in the type of embedding used: in [54] we used the encoding produced by Word2Vec and therefore we obtained a numeric vector for each word in a text document, while in this work we propose the textual features extracted from a CNN network for text documents classification, trained using all the text available in a document. In the next sections, the encoding technique used to graphically represent the text above the image, will be summarized.

Multimodal fusion methods are successfully employed to other modalities, e.g. video and audio [84, 25].

Other interesting examples of multimodal approaches that make use of deep networks include restricted Boltzmann machines [90], auto-encoders [91].

4.2 The Proposed Approach

In this work we take a cue from our previous work [54] to transform a text document into an image to be classified with a CNN. However, instead to use numeric values from Word2Vec model to represent a text document, we are using a new approach involving a CNN trained for text classification.

First, we transform the text document into a visual representation to construct an information enriched image containing text and image. Finally, we solve the multimodal problem using this new image to train a CNN generally used for image classification.

We use a variant of the CNN model proposed by Kim [47] for text document classification. The input layer is a text document followed by a convolution layer with multiple filters, then a max-pooling layer followed by a fully connected layer, and finally a softmax classifier. The network configuration summary is show in Table 4.1. Text features are extracted from the fully connected layer (Figure 4.1a) and transformed into an RGB encoding so that it can be overlaid onto an image associated with the text document.

We observed that if we concatenate the scaled and resized image (Figure 3.6b) to the text document features (Figure 4.1a), before passing at the output level, the model learns a better representation of the text features. Figure 4.1 shows architectural representation of the model used to encode the text dataset into an image dataset to obtain a multimodal dataset. In the second step, resulting images are fed to any baseline CNN for classification.

As an advantage of this method we can cast a single modality model into a multimodal model without the need of adapting the model itself. This approach is suitable to be adopted in multimodal methods because a CNN architecture can extract information from both the encoded text and the related image.

4.2.1 Encoding Scheme

We exploit the CNN model proposed by Kim [47] which performs text to visual features transformation within a single step. Figure 4.1 summarizes the encoding system used in this work, where a reshape was applied to the fully connected layer showed in Figure 4.1a

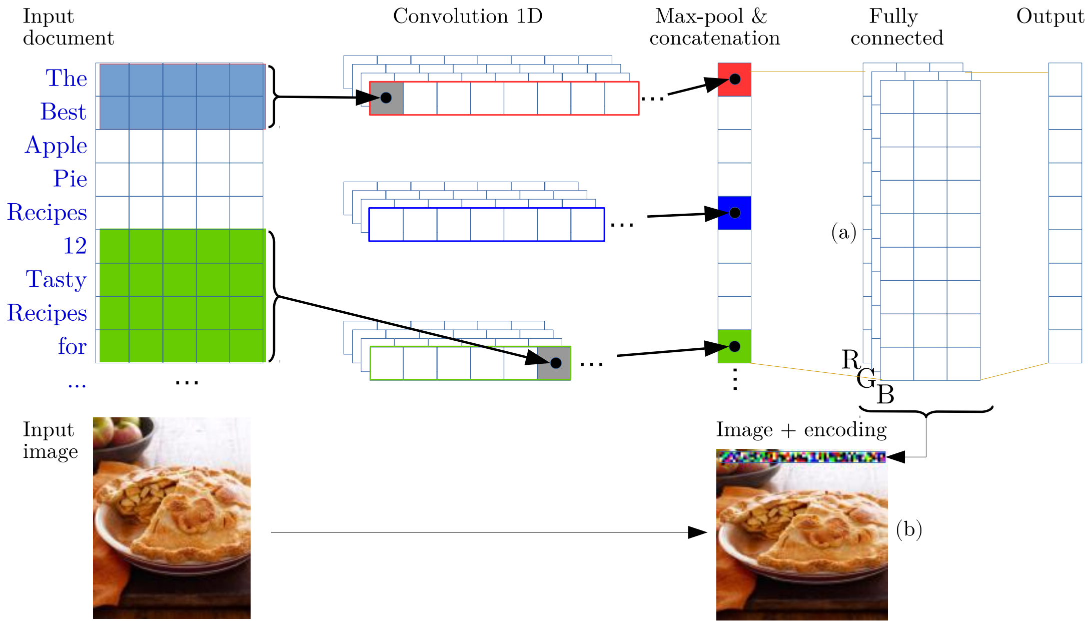


Figure 4.1: The proposed text and image fusion model for deep multi-modal classification. The text is encoded (a) is concatenated to the resized image (b) so that the output layer can consider simultaneously the text along with the information of the image. After the training step only the text features (a) are extracted to be drawn over the original image and thus generate a new multimodal dataset.

to transform an array into an image representing the encoded text to be superimposed on the original image.

Features are extracted from the trained CNN model and transformed into a visual representation of the document.

In practice, we used feature vectors showed in Figure 4.1(a), having a size $L = 3 \cdot w \cdot h$ that is a multiple of 3 in order to be transformed into a color image. We used the same concept of *superpixel* used in [54] to represent a sequence of three values $\in L$ as an area with a uniform color of $P \times P$ dimension. In this way textual features are represented as a sequence of superpixel, drawn from left to right and from top to bottom, starting from a certain position of the scaled image (see some examples of the final multimodal image in Figure 4.2 and Figure 4.3).

Finally, we encode an entire text document within the image plane and then the next multimodal CNN model can work simultaneously on both modalities.

This approach has an advantage to the work in [54], in fact in our work it is possible to encode long text documents because we encode the entire document in the same image area having fixed size equals to $w \times h \times 3$. Some examples of resulting images with encoded textual features can be seen in Figure 4.2.

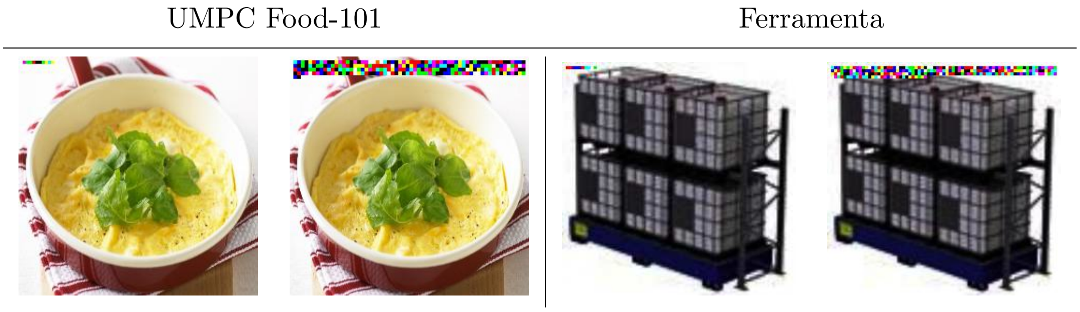


Figure 4.2: Two encoding examples taken from the two datasets. Images on the left column show an encoding of 10 superpixel length while on the right column we have an encoding of length equal to 250 superpixel. All images are 227 × 227 in size having an encoding superpixel equals to 3×3 pixels for Ferramenta dataset and 4×4 for Food-101 dataset.

Table 4.1: Network configuration summary. k, s and p stand for kernel size, stride and padding size respectively. In the convolution layer, we use 128 filters for each of the following sizes 3,4,5 (the first one is showed below). The embedding size is $w = 128$.

Type	Configuration
Output	num. classes
Fully Connected	H_t(encoded-text-height) $\times$ W(out-img-width) $\times 3$ + (H(out-img-height) $-H_t$)$\times$ W $\times$ 3
MaxPool-1D	h:(encoded-text-height), k:1, s:1, p:1
Convolutional-1D	w(embedding-size), k:$3 \times w$, s:1, p:1
Input	100 words (sequence length)

4.3 Experiments

4.3.1 Preprocessing

The proposed multimodal approach transforms text descriptions and embeds them onto associated images to obtain information enriched images. An example of information enriched image is shown in Figure 4.2. In this work, the transformed text description is embedded into a RGB image with an image size of 227 × 227 for UMPC Food-101 and Ferramenta multimodal dataset.

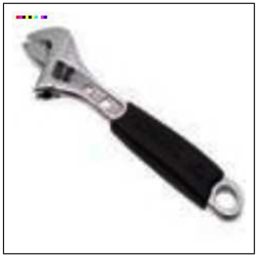
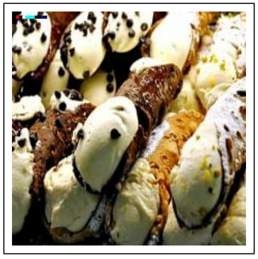
Ferramenta		UMPC Food-101	
	bahco 9070p chiave inglese regolabile ergonomica 15 3 cm 6 pollici a becco reversibile colore nero		Cannoli Recipe - Food.com
	connex cox550110 chiave inglese regolabile 25 4 cm		homemade cannoli filling The 350 Degree Oven
	axis 28831 chiave inglese regolabile con impugnatura morbida e rullo estremamente scorrevole 200 mm e 300 mm . . .		Cake Boss Cannoli Cake Ideas and Designs
	sam outillage 54 c10 chiave a rullino cromata 10 lunghezza 255 mm sam		Scones* Biscotti* Cannoli on Pinterest
	faithfull chiave regolabile 150 mm		Sicilian Cannoli Recipe The Daily Meal

Figure 4.3: Each column contains 5 images and associated text descriptions belonging to a particular class of Ferramenta and UPMC Food-101 datasets. Futhermore, each image contains the proposed encoded text. Note that the text encodings on each column are similar to each other even if the text and images are different from each other.

Table 4.2: Comparison of our approach with early and late fusion strategies. The results on the Ferramenta dataset were extracted from paper [53]

Dataset	Early F.	Late F.	Proposed
Ferramenta	89.53	94.42	**95.15**
Food-101	60.83	34.43	**82.90**

4.3.2 Detailed CNN settings

We used a standard AlexNet [31] and GoogleNet [44] on the Deep Learning GPU Training System (DIGITS) with default configuration. For fair comparison, we used same CNN settings for experiments using only images and fused images. We use standard CNN hyperparameters. The initial learning rate is set to 0.01 along with Stochastic Gradient Descent (SGD) as optimizer. The network is trained for a total of 60 epochs and/or till no further improvement is noticed to avoid over fitting. In our experiments, accuracy is used to measure classification performances. The aim of the experiment is to show that by adding encoded text information in images it is possible to obtain better classification results compared to the best one obtained using a single modality (Text/Image). We conducted following experiments with this aim in mind: (1) classification with CNN using only images, (2) classification with CNN using only text descriptions, (3) classification with CNN using fused images, (4) comparison with early and late fusion strategies.

The first experiment consists of extracting only text descriptions from multimodal datasets, then we train text classification model shown in Figure 4.1. Results are shown in first column of Table 4.3. It is very important to observe how the textual encoding extracted is very similar to each other when the text description represents very similar objects, even when the text information and the images are different from each other (see the example of text encoding showed in Figure 4.3).

The second experiment consists of extracting only images from multimodal datasets, then we train AlexNet [31] and GoogleNet [44] CNNs from scratch using DIGITS. Second and third columns of Table 4.3 shows these results. Images in Ferramenta multimodal dataset contain objects on a white background, this explains excellent classification results obtained on images alone. On the contrary, images in the UPMC Food-101 multimodal dataset are with complex background and extracted from different contexts, which leads to a low classification performance on images only.

The third experiment consists of employing fused images from multimodal datasets. We train AlexNet [31] and GoogleNet [44] CNNs from scratch using DIGITS.

Results in Table 4.3 indicate that the proposed fusion approach outperforms uni-modal methods. Furthermore, the approach is language independent, Ferramenta text descrip-

Table 4.3: Classification results comparison on only-text, only-image and fused images. There are two baseline models for images and fused-images, while we use only one baseline for text-only scores.

Dataset	Text	Image		Fusion	
		AlexNet	GoogleNet	AlexNet	GoogleNet
Ferramenta	92.09	92.36	92.47	95.15	**95.45**
Food-101	79.78	42.01	55.65	82.90	**83.37**

tions are in Italian language. Results on UPMC Food-101 clearly indicate benefit of our proposed approach, increasing the classification performance by two folds. This performance gain is due to leveraging on multimodal representation learning.

In fourth experiment, we compare our approach with early and late fusion as shown in Table 4.2. Experimental setting is inspired from the work [53]. In particular we used *Logarithmic Opinion Pool* [7] as a late fusion approach using Random Forest model applied to the 1000 BoW while as early fusion we used a SVM on the concatenation of Doc2Vec features and 4096 visual features from a trained CNN. Our proposed approach overcomes standard early and late fusion strategies which further reinforces strength of our approach.

The Figure 4.4 explores text embedding dimension sizes against two different CNN based architectures i.e. text only and fused image. We see that with lower text-embedding dimension, the fused architecture has an increased performance as compared to the text only architecture. Eventually, both architectures plateau as embedding dimension increases. However, the fused image architecture always maintains the upper bound over the other.

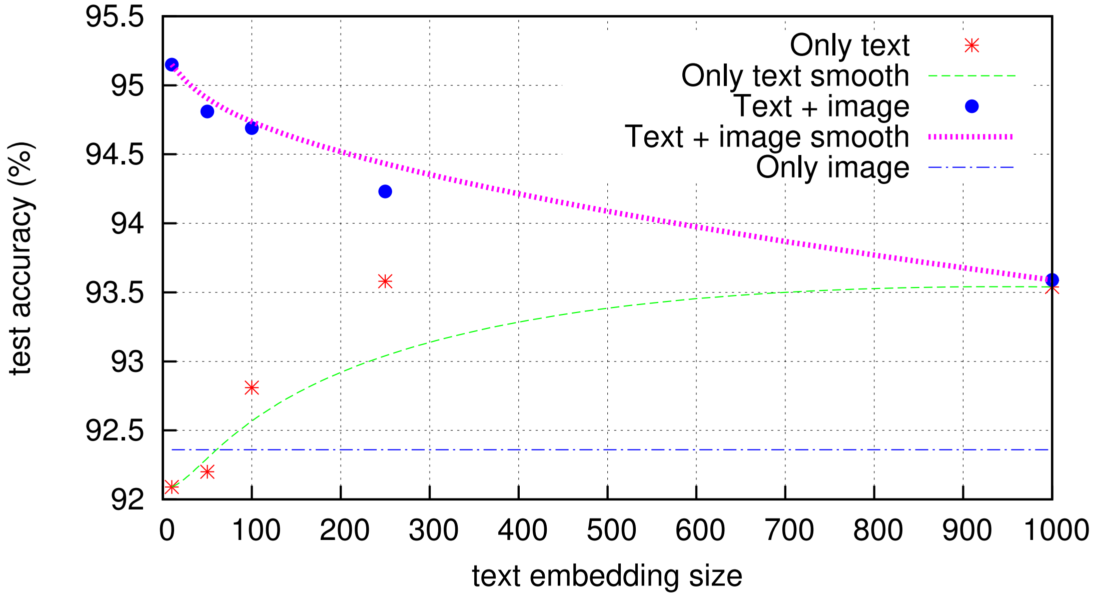


Figure 4.4: Comparison between the CNN that uses only text with the CNN that uses fusion of image and text, as the dimension of the text embedding varies. In this experiment the Ferramenta multimodal dataset is used.

5
GIT Loss for Deep Face Recognition

This chapter contains an overview of the paper ”GIT Loss for Deep Face Recognition” [21] presented at the British Machine Vision Conference (BMVC2018).

5.1 Introduction

The current decade is characterized by the widespread use of deep neural networks for different tasks [16, 15, 14]. Similarly, deep convolutional networks have brought about a revolution in face verification, clustering and recognition tasks [18, 14, 19, 17, 20]. Majority of face recognition methods based on deep convolutional networks (CNNs) differ along three primary attributes as explained in [17, 92]. The first is the availability of large scale datasets for training deep neural networks. Datasets such as VGGFace2 [93], CASIA-WebFace [94], UMDFaces [95], MegaFace [96] and MS-Celeb-1M [97] contain images ranging from thousands to millions. The second is the emergence of powerful and scalable network architectures such as Inception-ResNet [98] to train on large scale datasets. The last attribute is the development of loss functions to effectively modify inter and intra-class variations such as Contrastive loss [99], Triplet loss [18] and Center loss [19], given that softmax penalizes only the overall classification loss.

We employ all three attributes associated with face recognition. We use a large scale publicly available dataset, VGGFace2, to train the powerful Inception ResNet-V1 network. We propose a new loss function named Git loss to enhance the discriminative power of deeply learned face features. Specifically, the Git loss simultaneously minimizes intra-class variations and maximizes inter-class distances. A toy example that explains

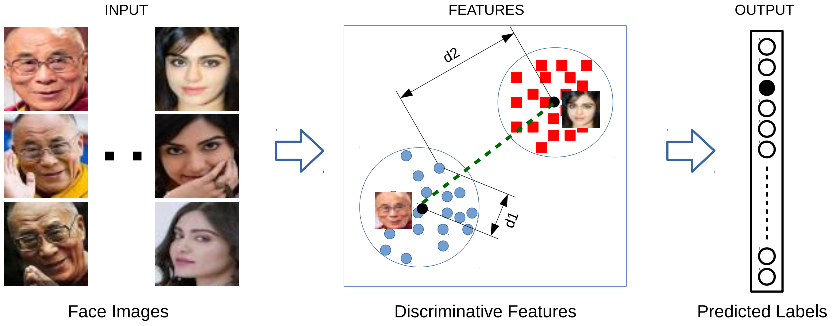


Figure 5.1: A toy example depicting the aim of our work: a CNN trained for face recognition supervised by our Git loss function that maximizes the distance $d2$ between features and centroids of different classes and minimizes the distance $d1$ between features and the centroid of the same class.

our approach is shown in Figure 5.1. The name of the loss function is inspired from two common Git version control software commands, "push" and "pull", which are semantically similar to the aim of this work: push away features of different identities while pulling together features belonging to the same identity.

In summary, main contributions of our paper include:

- A novel loss function which leverages on softmax and center loss to provide segregative abilities to deep architectures and enhance the discrimination of deep features to further improve the face recognition task
- Easy implementation of the proposed loss function with standard CNN architectures. Our network is end-to-end trainable and can be directly optimized by fairly standard optimizers such as Stochastic Gradient Descent (SGD).
- We validate our ideas and compare Git loss against different supervision signals. We evaluate the proposed loss function on available datasets, and demonstrate state-of-the-art results.

5.2 Related Work

Recent face recognition works are roughly divided into four major categories: (i) Deep metric learning methods, (ii) Angle-based loss functions, (iii) Imbalanced classes-aware loss functions and (iv) Joint supervision with Softmax. These methods have the aim of enhancing the discriminative power of the deeply learned face features. Deep learning methods [18, 100, 99] successfully employed triplet and contrastive loss functions for face recognition tasks. However, space and time complexities are higher due to the exponential growth of the datasets cardinality.

5.2.1 Deep Metric Learning Approaches

Deep metric learning methods focus on optimizing the similarity (contrastive loss [10, 11]) or relative similarity (triplet loss [12, 13]) of image pairs, while contrastive and triplet loss effectively enhance the discriminative power of deeply learned face features, we argue that both these methods can not constrain on each individual sample and require carefully designed pair and/or triplets. Thus, they suffer from dramatic data expansion while creating sample pairs and triplets from the training set with space complexity being $\mathcal{O}(n^3)$ for triplet networks..

5.2.2 Angle-based Loss Functions

Angular loss constrains the angle at the negative point of triplet triangles, leading to an angle and scale invariant method. In addition, this method is robust against the large variation of feature map in the dataset. ArcFace [92] maximizes decision boundary in angular space based on the L2 normalized weights and features.

5.2.3 Class Imbalance-Aware Loss Functions

Majority of loss functions do not penalize long tail distributions or imbalanced datasets. Range loss [101] employs the data points occurring in the long tail during the training process to get the k greatest ranges harmonic mean values in a class and the shortest inter-class distance in the batch. Although range loss effectively reduces kurtosis of the distribution, it requires intensive computation, hampering the convergence of the model. Furthermore, inter-class maximization is limited because a mini-batch contains only four identities. Similarly, center-invariant loss [20] handles imbalanced classes by selecting the center for each class to be representative, enforcing the model to treat each class equally regardless to the number of samples.

5.2.4 Joint Supervision with Softmax

In joint supervision with softmax based methods [102, 19], the discriminative power of the deeply learned face features is enhanced. The work in [19] penalizes the distance between deep features and their corresponding centers to enhance the discriminative ability of the deeply learned face features. With joint supervision of softmax loss and center loss function, inter-class dispersion and intra-class compactness is obtained. However, this comes with the cost of drastic memory consumption with the increase of CNN layers. Similarly, marginal loss [17] improves the discriminative ability of deep features by simultaneously minimizing the intra-class variances as well as maximizing the inter-class distances by focusing on marginal samples.

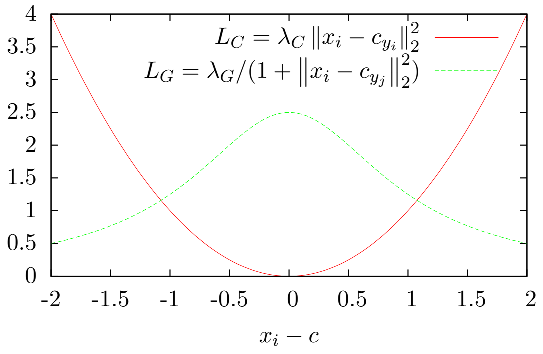


Figure 5.2: Graphical representation of L_C and L_G varying the distance $(x_i - c)$ in the range $[-2, 2]$. The L_G function takes a maximum value of λ_G at $x_i - c = 0$ and has an horizontal asymptote $L_G = 0$.

Inspired from two available works in the literature [19, 17], we propose a new loss function with joint supervision of softmax to simultaneously minimize the intra-class variations and maximize inter-class distances.

5.3 The Git Loss

In this paper, we propose a new loss function called Git loss inspired from the center loss function proposed in [19]. The center loss combines the minimization of the distance between the features of a class and their centroid with the softmax loss to improve the discriminating power of CNNs in face recognition.

In this work, to further improve the center loss function, we add a novel function that maximizes the distance between deeply learned features belonging to different classes (push) while keeping features of the same class compact (pull). The new Git loss function is described in Equation 5.1:

$$\begin{aligned} L &= L_S + \lambda_C L_C + \lambda_G L_G \\ &= -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda_C}{2} \sum_{i=1}^{n} \left\|x_i - c_{y_i}\right\|_2^2 + \lambda_G \sum_{i,j=1, i\neq j}^{m} \frac{1}{1 + \left\|x_i - c_{y_j}\right\|_2^2} \end{aligned} \tag{5.1}$$

where L_G is equal to $\frac{1}{1+\|x_i - c_{y_j}\|_2^2}$ which is responsible for maximizing the distance between divergent identities. The deep features of the i-th samples belonging to the y_i-th identity are denoted by $x_i \in \mathbb{R}^d$. The feature dimension d is set as 128, as reported in [18]. $W_j \in \mathbb{R}^d$ denotes the j-th column of the weights $W \in \mathbb{R}^{d\times n}$ in the last fully connected layer and $b \in \mathbb{R}^n$ is the bias term. c_{y_i} is the center of all deep features x_i

Figure 5.3: Some sample images taken from the VGGFace2 dataset aligned and cropped to 160 × 160 pixels.

belonging to the y_i-th identity. When the parameter $\lambda_G = 0$ the center loss function can be obtained.

The gradient ($\frac{\partial L_G}{\partial x_i}$) of the L_G with respect to x_i can be computed as:

$$\frac{\partial L_G}{\partial x_i} = \frac{\partial}{\partial x_i}(\frac{1}{1 + \|x_i - c_{y_j}\|_2^2}) \tag{5.2}$$

Let $u = 1 + ||x_i - c_{y_j}||_2^2$, thus $f = u^{-1}$. The equation 5.2 can be solved to compute the final gradient. We substitute the values of u and get $\frac{\partial u^{-1}}{\partial u} \frac{\partial}{\partial x_i}(1 + \|x_i - c_{y_j}\|_2^2)$. Solving $\frac{\partial u^{-1}}{\partial u}$ and $\frac{\partial}{\partial x_i}(1 + \|x_i - c_{y_j}\|_2^2)$, we get $\frac{-1}{u^2}$ and $2(x_i - c_{y_j})$ respectively. Substituting these values, the final equation becomes $\frac{-1}{u^2}(2(x_i - c_{y_j}))$. We can simplify this equation to obtain the final gradient equation 5.3.

$$= \frac{-2(x_i - c_{y_j})}{(1 + (x_i - c_{y_j})^2)^2} \tag{5.3}$$

Git loss simultaneously minimizes the intra-class variances using the L_C function and maximizes the inter-class distances using the L_G function. Parameters λ_C and λ_G are used to balance the two functions L_C and L_G respectively. From the plot of L_C and L_G functions, shown in Figure 5.2, it can be observed how these two functions have opposite behaviors: to minimize L_C we have to reduce the distance between features and the centers, while to maximize L_G we must maximize the distance between features and all centroids of other classes. L_G is a continuous and differentiable function, thus it can be used to train CNNs optimized by the standard Stochastic Gradient Descent (SGD) [103].

5.4 Experiments

We report experimental results on currently popular face recognition benchmark datasets, Labeled Faces in the Wild (LFW) [51] and YouTube Faces (YTF) [52]. We also report a set of experiments to investigate the hyper-parameters associated with the loss function.

5.4.1 Experimental Settings

Training Data.

We use data from VGGFace2 [93] and MS-Celeb-1M [97] dataset to train our model. The VGGFace2 dataset contains 3.31 million images of $9,131$ identities, with an average of 362.6 images for each identity. Moreover, the dataset is characterized by a large range of poses, face alignments, ages and ethnicities. The dataset is split into train and test sets with $8,631$ and 500 identities respectively, but we only used the train set. Some representative images taken from the VGGFace2 dataset are shown in Figure 5.3. MS-Celeb-1M dataset contains about 10M images for 100K celebrities. The dataset incorporates diversity in terms of professions, age, and regions. To report fair results, we remove overlapping identities with YTF and LFW datasets from the training set.

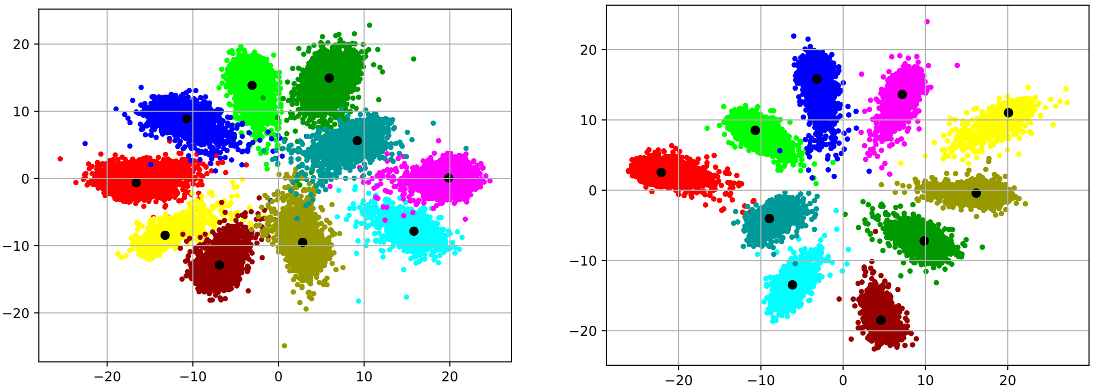
Data Preprocessing.

The label noise is minimized through automated and manual filtering in the VGGFace2 dataset. We applied horizontal flipping and random cropping data augmentation techniques to images, then face images are aligned using the Multi-Task CNN [104] and finally cropped to a size of 160×160 pixels before feeding to the network. We noticed that VGGFace2 contains 496 overlapping identities with LFW and 205 with YTF datasets, therefore, we removed overlapping identities from both datasets to report fair results. Similarly, MS-Celeb-1M dataset contains about 3184 overlapping identities with LFW and about 1077 with YTF datasets.

Network Settings.

We implemented the proposed Git loss in Tensorflow [105] and the network was trained using Nvidia's GeForce GTX 1080 GPU. The implementation is inspired from the *facenet* work, available on Github[1]. We employ the Inception ResNet-V1 network architecture and process 90 images in a batch. We use adaptive learning rate for the training process with a starting value of -1 and decreased it by a factor of 10 with Adam Optimizer [106], thus adding robustness to noisy gradient information and various data modalities across the dataset, improving the performance of the final model.

[1] https://github.com/davidsandberg/facenet



$\lambda_C = 0.01 \quad \lambda_G = 0$

$\lambda_C = 0.01 \quad \lambda_G = 0.1$

Figure 5.4: Two plots showing the behavior of Center loss (a) and Git loss (b) on MNIST training set. Using the Git loss function features are more compact (smaller intra-class distances) and more spaced (larger inter-class distances), further enhancing the discriminative power of deep features. Points with different colors denote features from different classes.

Test Settings.

Deep face features ($128 - dimensional$) are taken from the output of the fully connected layer. Since the features are projected to Euclidean space, the score is computed using the Euclidean distance between two deep features. Threshold comparison is obtained with $10 - fold$ cross validation for verification task. We employ two different trained models for LFW and YTF datasets due to number of overlapping identities with VGGFace2 dataset.

5.4.2 Experiments with λ_C and λ_G parameters

Parameters λ_C and λ_G are used to balance two loss functions L_C and L_G with softmax. In our model, λ_C controls intra-class variance while λ_G controls inter-class distances. We conducted various experiments to investigate the sensitiveness of these two parameters. These tests are systematic random search heuristics based. The major reason for employing heuristic methodologies over techniques like GridSearch is that when the dimensionality is high, the number of combinations to search becomes enormous and thus techniques like GridSearch become an overhead. The work in [107] argues why performance of GridSearch is not satisfactory as compared to other techniques. Table 5.1 shows average result values over 10 runs on MNIST dataset, we have following outcomes: (i) Smaller values of λ_C increase inter-class distance, but they also increase intra-class distances which is undesirable in face recognition. (ii) Our loss function produces higher

Table 5.1: Comparison between Center loss (λ_C) and Git loss (λ_G) on MNIST dataset. Values are obtained by averaging 10 runs. We highlighted best results compared to Center loss ($\lambda_G = 0$) for each configuration. We achieved reduced intra-class distance along with higher inter-class distance compared to Center loss.

λ_C	λ_G	**Loss**	**Train Acc.(%)**	**Val. Acc.(%)**	**Inter Dist.**	**Intra Dist.**
0.0001	0	0.020	99.77	98.52	85.95	8.39
0.0001	0.0001	0.0132	100.00	98.65	87.87	8.52
0.0001	**0.001**	**0.016**	**99.77**	**98.66**	**89.82**	**8.20**
0.0001	0.01	0.020	99.77	98.62	88.48	8.54
0.0001	0.1	0.032	99.61	98.46	96.76	9.56
0.0001	1	0.466	89.77	88.95	137.37	15.45
0.0001	1.5	0.641	80.63	79.56	160.28	16.91
0.0001	2	1.001	69.84	68.51	125.84	17.78
0.001	0	0.021	99.61	98.67	44.36	3.10
0.001	0.0001	0.117	96.75	95.66	51.77	4.75
0.001	0.001	0.024	99.84	98.53	47.81	3.23
0.001	**0.01**	**0.025**	**99.77**	**98.62**	**46.13**	**3.16**
0.001	0.1	0.053	99.22	98.69	51.22	3.46
0.001	1	0.779	76.10	76.33	68.77	6.55
0.001	1.5	0.460	89.22	89.06	89.67	7.45
0.001	2	0.757	80.94	81.89	96.02	9.14
0.01	0	0.025	99.65	98.89	21.36	1.09
0.01	0.0001	0.031	99.69	98.77	22.07	1.16
0.01	0.001	0.024	100.00	98.75	20.63	1.18
0.01	0.01	0.051	99.53	98.61	21.96	1.09
0.01	**0.1**	**0.037**	**99.84**	**98.70**	**22.93**	**1.22**
0.01	1	0.937	71.17	71.38	30.25	2.02
0.01	1.5	0.368	97.58	96.50	50.56	3.10
0.01	2	0.824	83.44	84.10	46.35	3.03
0.1	0	0.040	99.74	98.89	9.76	0.38
0.1	0.0001	0.049	99.53	98.85	9.65	0.42
0.1	0.001	0.024	100.00	98.96	10.33	0.38
0.1	0.01	0.026	99.92	99.00	9.76	0.37
0.1	**0.1**	**0.040**	**100.00**	**98.96**	**10.99**	**0.37**
0.1	1	1.508	57.11	57.90	10.52	0.55
0.1	1.5	1.741	53.59	54.03	10.81	1.03
0.1	2	1.536	67.98	66.65	15.43	1.03
1	0	0.031	100.00	99.00	5.12	0.14
1	0.0001	0.178	96.72	95.72	5.86	0.22
1	0.001	0.023	100.00	99.03	4.94	0.12
1	0.01	0.027	100.00	99.04	5.03	0.12
1	0.1	0.064	99.92	99.04	5.25	0.14
1	**1**	**0.264**	**99.92**	**99.02**	**8.30**	**0.21**
1	1.5	0.330	100.00	98.96	9.73	0.23
1	2	0.847	91.10	89.79	9.22	0.25

inter-class distances and lower intra-class distances. An example displaying the qualitative and quantitative results of our Git loss and Center loss is shown in Figure 5.4. Note that these results are obtained with a single run on MNIST dataset.

5.4.3 Experiments on LFW and YTF datasets

We compare the Git loss against many existing state-of-the-art face recognition methods in Table 5.2. From results, we can see that the proposed Git loss outperforms the softmax loss by a significant margin, from 98.40% to 99.30% in LFW and from 93.60% to 95.30% in YTF. In addition, we compare our results with center loss method using the same network architecture (Inception-ResNet V1) and dataset (VGGFace2). Furthermore, we also compare our results with center loss by training on MS-Celeb-1M dataset with same baseline architecture (Inception-ResNet V1). The Git loss outperforms the center loss, obtaining an accuracy of 99.30% as compared to 99.20% on LFW and 95.30% compared to 95.10% on YTF. These results indicate that the proposed Git loss further enhances the discriminative power of deeply learned face features. Moreover, we trained our model with $\approx$ 3M images which are far less than other state-of-the-art methods such as [18, 108, 109], reported in Table 5.2.

Methods	Images	LFW(%)	YTF(%)
DeepID [99]	-	99.47	93.20
VGG Face [14]	2.6M	98.95	97.30
Deep Face [109]	4M	98.37	91.40
Fusion [108]	500M	98.37	-
FaceNet [18]	200M	99.63	95.10
Baidu [100]	1.3M	99.13	-
Range Loss [110]	1.5M	99.52	93.70
Multibatch [111]	2.6M	98.80	-
Aug [112]	0.5M	98.06	-
Center Loss [19]	0.7M	99.28	94.90
Marginal Loss [17]	4M	99.48	95.98
Softmax	$\approx$ 3M	98.40	93.60
Center Loss [19]	$\approx$ 3M	99.20	95.10
Git Loss (Ours)	$\approx$ 3M	**99.30**	**95.30**
Center Loss [19]	10M	–	–
Git Loss (Ours)	10M	–	–

Table 5.2: Performance verification of different state-of-the-art methods on LFW and YTF datasets. The last three rows show results using the same architecture (Inception-ResNet V1) trained on the VGGFace2 dataset. We also compare centerloss and Git loss by training on MS-Celeb-1M dataset.

6 Conclusions and Future Works

6.1 Conclusion

In this thesis a new methodology that exploits both textual and visual information for document classification has been proposed. In addition to the introduction of a novel model able to merge textual and visual information within a single image, we introduced a novel loss function which is capable of minimizing intra-class distance, while enhancing inter-class distances between samples of datasets.

Document classification is not a trivial task and can face many obstacles (as described in Chapter 1) due to the shortness of the text associated to a document or due to inconsistencies between texts and images. This scenario is very common especially on e-commerce websites or social media platforms, where content is created by people without following rigorously grammar rules.

In literature, the problem of document classification has been tackled several times using very different strategies, such as, single-modal classification and Early or Late Fusion approaches. Simpler approaches like single modal classification do not take advantages of entire available information, like text, image and audio or video signals. In fact, often, very different elements appear to be very similar if considering only the textual description or the image, making the classification process error-prone. Exploitation of visual and textual information is used to help classifiers to make correct predictions, which, obviously cannot be performed using only one modality. This is a crucial aspect to keep in mind while trying to overcome data inconsistencies with automatic approaches. Fusion approaches can be used for multimodal classification, however, if considering the Early and Late Fusion approaches, many classifiers need to be trained making the pipeline computationally expensive.Moreover, the Early Fusion approach, described in Chapter 2 did not overcome results obtained with Late Fusion strategy.

The first objective of this thesis was to create a novel method of including textual and visual features onto a single image, being able to exploit capabilities of Deep Convolutional Neural Networks, which achieved state-of-the-art in the Computer Vision research field, on text classification.

The approach is widely described in Chapter 4. Summarizing, our proposal consists in using a single CNN architecture to add textual features on images. The network computes embeddings for text and, to get relevant features, applies 1-d convolutional kernels with different heights. This is similar to the application of *n-grams* strategy and allows to the model itself to select best feature extracted with different resolutions. Next, obtained embeddings features are encoded in the upper part of the images. The result is a set of images, containing feature from both modalities, which are then used as dataset to train another network which performs the classification task. Experimental results show that overall accuracy of a Convolutional Neural Network trained on these artificial images is considerably higher than accuracy obtained using only textual or visual information, especially when different documents have similar text or similar images.

The second objective of this thesis was to introduce a loss function which is capable of improving the discrimination between features. This task, known as Discriminative Feature Learning, is a well studied topic in literature because separability is the base condition to obtain good performance with any classifier. Many methods have been introduced in recent times, however, most of them require the creation of a dataset of pairs or triplets to train models in order to achieve minimum intra-class distance and maximum inter-class distance. Other approaches which try to obtain such similar result dealing with imbalanced datasets are available. The disadvantage of both kind of approaches is due to the high memory consumption and high computation time required, making their usage on large scale datasets very difficult.

In this thesis, we introduced a new loss function, named Git loss, described in Chapter 5, which makes deep feature more discriminable. We exploit the softmax as supervision signal and the well-known property of the center loss which compacts patterns of a class, lowering intra-class distances. The result is a novel loss function which minimizes intra-class variations and enlarges inter-class distances simultaneously. We experimented our method on two famous face recognition datasets, LFW and YTF, showing that classification and generalization abilities are improved.

6.2 Future Works

Future extensions of multimodal document classification work may rely on testing on large scale datasets with thousands of classes. The approach we built was born after considering that in many cases, the exploitation of only textual or visual features is not enough to provide a correct classification of a document. There are documents with similar images and very different text and vice-versa, thus, using a single source classification can lead to very poor performances. The first extension we will implement is to encode textual information not only in the upper part of the image, but changing pixel values of the whole image.

Moreover, we would like to include in the pipeline shown in Chapter 4, the text detection and recognition approach shown in Chapter 2. The advantage would be the inclusion of keyword that can be included into an image in the form of text which can enrich the textual component of our approach.

The discriminative feature learning approach, explained in Chapter 5, actually, has been developed and tested only to face datasets, however, it can be easily applied to any other topic. Thanks to the fact we implemented it in an end-to-end manner using Stochastic Gradient Descent as optimizer, it is very easy to use and to adapt to different datasets, thus it will be applied also to datasets used for multimodal Document classification.

All works I carried out during my three years Ph.D. were based on multimodal classification and discriminative feature learning. The final work ”Image and Encoded Text Fusion for Multimodal Classification” [9], can be viewed as an example of real application scenario and it is not limited only to images and texts, but can be exploited, for example, also for sounds. Text and images are associated to plenty of contexts, for example in most of social networks or marketplaces such as Facebook, Amazon, Twitter, so this approach can perfectly suit actual needs of companies.

Taking all this into consideration, information extraction and fusion inside images represents an effective approach useful for many research areas such as Sentiment Analysis, Image and Text Retrieval, Image Ranking.

As a proof of concept, we dedicated part of the time to explore above-mentioned fields achieving encouraging results with images obtained from described approaches.

Colophon

- This thesis was written using LaTeX.
- The LaTeX template for this thesis was made by Carullo Moreno.
- The code for experiments was developed in Python[1] and Java[2].
- Algorithms presented were developed using the Google TensorFlow[3], Pandas[4] and Scikit-learn[5] libraries.
- For the approach presented in Section 3.1 WEKA[6] for Java was used.
- NVIDIA DIGITS[7] was used to create, train and evaluate the model shown in Chapter 4.
- 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.python.org

[2] https://www.java.com/en/

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

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

[5] https://scikit-learn.org/stable/index.html

[6] https://www.cs.waikato.ac.nz/ml/weka/

[7] https://developer.nvidia.com/digits

Bibliography

[1] A. Ritter, S. Clark, O. Etzioni *et al.*, “Named entity recognition in tweets: an experimental study,” in *Proceedings of the conference on empirical methods in natural language processing.* Association for Computational Linguistics, 2011, pp. 1524–1534.

[2] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, “New avenues in opinion mining and sentiment analysis,” *IEEE Intelligent systems*, vol. 28, no. 2, pp. 15–21, 2013.

[3] C. T. Duong, R. Lebret, and K. Aberer, “Multimodal classification for analysing social media,” *Computing Research Repository, (CoRR)*, 2017.

[4] D. Zhang, Y. Wang, L. Zhou, H. Yuan, D. Shen, A. D. N. Initiative *et al.*, “Multimodal classification of alzheimer’s disease and mild cognitive impairment,” *Neuroimage*, vol. 55, no. 3, pp. 856–867, 2011.

[5] V. Kalidas and L. Tamil, “Cardiac arrhythmia classification using multi-modal signal analysis,” *Physiological measurement*, vol. 37, no. 8, p. 1253, 2016.

[6] D. Lu and Q. Weng, “A survey of image classification methods and techniques for improving classification performance,” *International journal of Remote sensing*, vol. 28, no. 5, pp. 823–870, 2007.

[7] L. Rokach, “Ensemble-based classifiers,” *Artificial Intelligence Review*, vol. 33, no. 1, pp. 1–39, 2010.

[8] D. Opitz and R. Maclin, “Popular ensemble methods: An empirical study,” *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.

[9] I. Gallo, A. Calefati, S. Nawaz, and M. K. Janjua, “Image and encoded text fusion for multi-modal classification,” in *2018 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Dec 2018, pp. 1–7.

[10] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 539–546.

[11] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *null*. IEEE, 2006, pp. 1735–1742.

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

[13] 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.

[14] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[15] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *CVPR*, vol. 3, no. 6, 2017, p. 7.

[16] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua, "Neural aggregation network for video face recognition." in *CVPR*, vol. 4, no. 6, 2017, p. 7.

[17] 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.

[18] 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.

[19] 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.

[20] Y. Wu, H. Liu, J. Li, and Y. Fu, "Deep face recognition with center invariant loss," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. ACM, pp. 408–414.

[21] A. Calefati, M. Janjua, S. Nawaz, and I. Gallo, "Git loss for deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[22] 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.

[23] A. Fleury, M. Vacher, and N. Noury, "Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *IEEE transactions on information technology in biomedicine*, vol. 14, no. 2, pp. 274–283, 2010.

[24] L. Gómez-Chova, D. Tuia, G. Moser, and G. Camps-Valls, "Multimodal classification of remote sensing images: A review and future directions," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1560–1584, 2015.

[25] 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.

[26] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Machine learning: ECML-98*, pp. 137–142, 1998.

[27] M. Rogati and Y. Yang, "High-performing feature selection for text classification," in *Proceedings of the eleventh international conference on Information and knowledge management.* ACM, 2002, pp. 659–661.

[28] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.

[29] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[30] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] 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.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *international Conference on computer vision & Pattern Recognition (CVPR'05)*, vol. 1. IEEE Computer Society, 2005, pp. 886–893.

[33] Y. Bai, L. Guo, L. Jin, and Q. Huang, "A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition," in *2009 16th IEEE International Conference on Image Processing (ICIP).* IEEE, 2009, pp. 3305–3308.

[34] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for svms," in *Advances in neural information processing systems*, 2001, pp. 668–674.

[35] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "A simultaneous feature adaptation and feature selection method for content-based image retrieval systems," *Knowledge-Based Systems*, vol. 39, pp. 85–94, 2013.

[36] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International conference on artificial neural networks.* Springer, 1997, pp. 583–588.

[37] J. Li and R. R. Linear, "Principal component analysis," 2014.

[38] 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.

[39] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[40] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *Computing Research Repository, (CoRR)*, 2015.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[42] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *International Conference on Learning Representations (ICLR)*, 2014.

[43] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *Computing Research Repository, (CoRR)*, 2013.

[44] 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.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computing Research Repository, (CoRR)*, 2014.

[46] 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.

[47] Y. Kim, "Convolutional neural networks for sentence classification," *Computing Research Repository, (CoRR)*, 2014.

[48] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *Computing Research Repository, (CoRR)*, 2014.

[49] ——, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Advances in neural information processing systems*, 2015, pp. 919–927.

[50] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *Computing Research Repository, (CoRR)*, 2015.

[51] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[52] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE, 2011, pp. 529–534.

[53] I. Gallo, A. Calefati, and S. Nawaz, "Multimodal classification fusion in real-world scenarios," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 05, Nov 2017, pp. 36–41.

[54] I. Gallo, S. Nawaz, and A. Calefati, "Semantic text encoding for text classification using convolutional neural networks," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 05, Nov 2017, pp. 16–21.

[55] A. C. I. Gallo and S. Nawaz, "Reading numbers in the wild," *Submitted to CAIP2019.*

[56] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* IEEE, 2010, pp. 902–909.

[57] Y. Feng and M. Lapata, “Visual information in semantic representation,” in *Annual Conference of the North American Chapter of the Association for Computational Linguistics.* ACL, 2010, pp. 91–99.

[58] M. J. Huiskes, B. Thomee, and M. S. Lew, “New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative,” in *Proceedings of the international conference on Multimedia information retrieval.* ACM, 2010, pp. 527–536.

[59] C. W. Leong and R. Mihalcea, “Going beyond text: A hybrid image-text approach for measuring word relatedness.” in *IJCNLP*, 2011, pp. 1403–1407.

[60] 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.

[61] 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.

[62] 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.

[63] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *Computing Research Repository, (CoRR)*, vol. abs/1301.3781, 2013.

[64] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[65] Y. Peng, X. Zhou, D. Z. Wang, I. Patwa, D. Gong, and C. V. Fang, “Multimodal ensemble fusion for disambiguation and retrieval,” *IEEE MultiMedia*, vol. 23, no. 2, pp. 42–52, 2016.

[66] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[67] P. Blunsom, E. Grefenstette, and N. Kalchbrenner, “A convolutional neural network for modelling sentences,” in *Proceedings of the 52nd Annual Meeting of the*

Association for Computational Linguistics. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[68] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification." in *ACL (1)*, 2014, pp. 1555–1565.

[69] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[70] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[71] L. Gómez, M. Rusinol, and D. Karatzas, "Cutting sayre's knot: Reading scene text without segmentation. application to utility meters," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS).* IEEE, 2018, pp. 97–102.

[72] G. Li, S. Xu, X. Liu, L. Li, and C. Wang, "Jersey number recognition with semi-supervised spatial transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1783–1790.

[73] M. Bušta, L. Neumann, and J. Matas, "Deep textspotter: An end-to-end trainable scene text localization and recognition framework," in *Computer Vision (ICCV), 2017 IEEE International Conference on.* IEEE, 2017, pp. 2223–2231.

[74] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4168–4176.

[75] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 970–983, 2014.

[76] B. Yu and H. Wan, "Chinese text detection and recognition in natural scene using hog and svm," *DEStech Transactions on Computer Science and Engineering*, 2016.

[77] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, "T-hog: An effective gradient-based descriptor for single line text regions," *Pattern recognition*, vol. 46, no. 3, pp. 1078–1090, 2013.

[78] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[79] I. Gallo, A. Zamberletti, and L. Noce, "Robust angle invariant gas meter reading," in *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on.* IEEE, 2015, pp. 1–7.

[80] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Asian Conference on Computer Vision.* Springer, 2010, pp. 770–783.

[81] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[82] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[83] D. Kiela, E. Grave, A. Joulin, and T. Mikolov, "Efficient large-scale multi-modal classification," *Proceedings of AAAI 2018*, 2018.

[84] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *International Conference on Machine Learning (ICML)*, 2011, pp. 689–696.

[85] 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.

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

[87] 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.

[88] E. Shutova, D. Kiela, and J. Maillard, "Black holes and white rabbits: Metaphor identification with visual features," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 160–170.

[89] G. Evangelopoulos, A. Zlatintsi, A. Potamianos, P. Maragos, K. Rapantzikos, G. Skoumas, and Y. Avrithis, "Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention," *IEEE Transactions on Multimedia*, vol. 15, no. 7, pp. 1553–1568, 2013.

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

[91] P. Wu, S. C. Hoi, H. Xia, P. Zhao, D. Wang, and C. Miao, "Online multimodal deep similarity learning with application to image retrieval," in *ACM international conference on Multimedia*. ACM, 2013, pp. 153–162.

[92] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *arXiv preprint arXiv:1801.07698*, 2018.

[93] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," *arXiv preprint arXiv:1710.08092*, 2017.

[94] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[95] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa, "Umdfaces: An annotated face dataset for training deep networks," in *Biometrics (IJCB), 2017 IEEE International Joint Conference on*. IEEE, 2017, pp. 464–473.

[96] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4873–4882.

[97] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 87–102.

[98] 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.

[99] 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.

[100] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.

[101] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 5419–5428.

[102] Y. Sun, X. Wang, and X. Tang, "Sparsifying neural network connections for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4856–4864.

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

[104] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[105] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[106] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[107] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[108] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Web-scale training for face identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2746–2754.

[109] ——, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[110] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5409–5418.

[111] O. Tadmor, Y. Wexler, T. Rosenwein, S. Shalev-Shwartz, and A. Shashua, "Learning a metric embedding for face recognition using the multibatch method," *arXiv preprint arXiv:1605.07270*, 2016.

[112] I. Masi, A. T. Trán, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?" in *European Conference on Computer Vision.* Springer, 2016, pp. 579–596.