Image Sentiment Analysis: Experimental Evaluation of Several Deep Learning Architectures

António P. Gaspar
antonio.pedro.gaspar@ubi.pt
Luís A. Alexandre
luis.alexandre@ubi.pt

Universidade da Beira Interior
Instituto de Telecomunicações
Rua Marquês d'Ávila e Bolama
6201-001, Covilhã, Portugal

Abstract

Image sentiment analysis is an important topic nowadays. It is possible to use it to classify an image at sentiment level, as negative, neutral or positive. However, to classify an image at this level is a hard challenge because its semantic meaning can represent many scenarios. In this paper, we present an analysis of several image classification methods that we evaluate to improve the state of the art in a large tweet data set.

1 Introduction

Sentiment analysis is a very studied subject. Currently, the data of social media networks is growing at each second, which makes them a good place to collect images. These can be analysed and classified for different purposes, such as sentiment analysis. There are methods capable to handle this job. However, the majority of these methods has a high text dependency to realise the sentiment classification. A widely used expression is "A picture is worth a thousand words", that means, that an image can transmit a large message. Nonetheless, the message is not always clear, and this situation can bring different interpretations, especially when the interpreters have different cultures. With this work, we propose an improvement of the method proposed in [4], to do sentiment analysis using only the images present in tweets. Section 2 presents related work, including the method from [4], that we partially improve in this paper. Section 3 describes our work. Section 4 contains the experiments and the final section has our conclusions.

2 Related Work

2.1 Sentiments

According to the psychology area, sentiments are different from emotions. This fact is described by the author of the paper [5]. Sentiments are the result of subjective experiences that were lived from an emotion. Emotions are the triggers for actions that can be positive or negative and are the base of sentiments. These can construct the history of the all feelings that are processed and memorised. This fact is important in sentiment analysis because through it is possible to reduce the subjectivity according to the culture where the analysed data belongs to. However, often the data cannot be organised by the culture. It is the case of the data collected from social media networks. For this reason, artificial intelligence may help to find the best features for classification. Next, we present some of the techniques related with the present theme.

2.2 Sentiments and Artificial Intelligence

Nowadays text, images, videos and all multimedia content can be processed and analysed. To analyse the data, most models represent information using sets of features which in turn represent the classes of the target objects. This process can be done through many different approaches, but currently, deep neural networks, such as Convolutional Neural Networks (CNNs), have been producing very good results when applied to image data.

There have been many proposals of methods for Image Sentiment Analysis. The authors of [7] studied the sentiment analysis process. They propose a method that is capable of classifying images at sentiment polarity level. The dataset they use is composed of 3 million tweets, which include text and images, and was constructed by them collecting the information on Twitter. For the classification, they propose a method that leverages the text classification and correlates it with the image. They conclude that text associated with image is often noisy and is weakly correlated with the image content, but it is possible to classify its sentiment using a model that is trained with the images classified with text labels.

In another work described in [2], the authors explore four different architectures of convolutional neural networks to do sentiment analysis in visual media. This work was based on a labelled set that has the main categories of the description of the scene. With their results, the authors compose their own dataset and train a model that improves the results. With this knowledge, following section present the method developed in [4].

2.3 Image Content Analysis

In [4] an image content analysis method was developed. This is a complex subject because an image might contain many objects. This work tries to identify automatically the class of the object that an image can represent. To do this a pre-trained model with the ImageNet [3] was used to classify the data into its class through the ImageNet classifications (1000 possible object classes). All images on the ImageNet are quality-controlled and human-annotated. An InceptionResNetV2 trained model was used, which according to the author [1], has 80.17% of accuracy. This model comes from a python package that is called pretrainedmodels [1]. In [4] the image content analysis was used to to build a probability distribution that made possible to classify an image according to its sentiment polarity, (negative, neutral or positive). So, with the InceptionResNetV2, a model was built that is fed with the union of the training and validation sets to increase the number of the images. The InceptionResNetV2 classified the contents of each dataset image into one of the ImageNet classes. Each of these images contains a sentiment classification in the training and validation sets that were used to build a table with the probability distribution of the image sentiment for each ImageNet class. This was then combined with text and image sentiment results to obtain the final classification.

3 Proposed Method

The method in [4] fused information from 3 different sources. In this paper we explore alternative network architectures that can improve the results obtained from the analysis of sentiment on isolated images.

3.1 Image Classification

The developed method for image analysis is based on a deep learning approach. This is implemented with Pytorch [6], which is a deep learning framework that supports several features and automatic differentiation. For this work, we explore three versions of the Resnet, which are, the ResNet18, the ResNet50, and the ResNet152. We explore too other architectures, the Inception V3 and DenseNet. We use these typologies because of they are the state-of-the-art methods that can reduce significantly the vanishing gradient problem. To use these models we need to set them up and prepare the data. To do that we follow the next steps.

3.2 Data Preprocessing

One of the biggest challenges in deep learning approaches is data quality. Any deep learning approach is hungry for data because it is through it that the network extracts and learns the features used for classification. The dataset used has many images with different scales and sizes. This fact can slow the training process. The pre-processing method used resizes each image to 224x224 in the case of the ResNets architectures, and 299x299 in the others.

This work was partially supported by Instituto de Telecomunicações under grant UID/EEA/50008/2019 and by project MOVES - Monitoring Virtual Crowds in Smart Cities (PTDC/EEI-AUT/28918/2017) financed by FCT - Fundação para a Ciência e a Tecnologia.

Table 1: Table of the experiments on different neural network architectures.

		Network Architectures				
		ResNet 18	ResNet 50	ResNet 152	InceptionV3	DENSENet161
Train	Time(H)	6	17	41	25	78
Test	LOSS	1.0490	0.9909	0.9821	0.9770	0.9730
	ACC	0.4474	0.5251	0.5234	0.5156	0.5274

3.3 Model Choice and its Adaptation

Pytorch comes with several built-in models. In this work, we selected five of these models, the ResNet18, ResNet50, ResNet152, InceptionV3, and DenseNet. All the models are set up with the same hyper-parameters. These are, the learning rate that starts with 0.001, the momentum with 0.9 and the gamma parameter with 0.1. We use an optimiser that will hold the current state and will update the parameters based on the computed gradients. This is the SGD (Standard Gradient Descendent). We use a schedule that provides several methods to adjust the learning rate based on the number of epochs. This will adjust the learning rate in every seven epochs. We define 30 epochs to train.

4 Experiments

In this section, we present the dataset used to evaluate our method, as well as the results of the experimental evaluation. The results that we present, were obtained on the test set. Table 3 presents a confusion matrix that shows some selected samples that can characterise our challenge, which is the subjectivity implicit on the images. The diagonal images represent correct predictions. The other cases show different types of error that the method makes. In these results we can see that the last image in the first row, which is truly negative was classified as positive by our method. Also the first image of the second row, in the neutral case is classified by our method as negative. In positive case, the second image in the last row, is classified as neutral. These cases occur because the dataset has a large homogeneity, in fact these situations occur in the real world, and the correct sentiment is assigned to the images by the context in which they are inserted.

4.1 The Dataset

The authors of [7] built a dataset with three million tweets. These tweets contain text and images. Nonetheless this huge amount of data, it has some problems, such as duplicate entries and malformed images. These situations led the authors to build a subset that is composed by tweets that have images and text in their corpus, non duplicated and non-malformed images, as well the same number of occurrences on the different classes. The subset is called *B-T4SA*, and is divided into three partitions: the train part, the validation part and the test part. Each one of these subsets has three classes, negative, neutral and positive. Each class has the same number of images as the others.

4.2 Results

We train the models with a GeForce GTX 1080 TI, using the training set to train and the validation set to validate the training phase. ResNet18 uses 512 features from each image and achieves 44.7%, ResNet50 and ResNet152 use 2048 features and achieve better results. ResNet50 achieves the best result, 52.5%, and exceeds the result presented in the dataset paper [7] for only the image analysis, which is 51.3%. However our intent is to improve this result. With DenseNet we can improve it. The DenseNet uses 2208 features, and the result which we obtained is 52.7% accuracy on the test set. Table 1 shows the experiments of the different used architectures. We can conclude that, the use of a Densenet 161 network improves the results slightly when compared to the other approaches. However when we analyse the time, which each network takes, in the training phase the ResNet50 take less time and achieved a closer result to the Densenet 161 that take 4 times more time to get a slightly better result. Despite these difference, Densenet 161 achieves a better result that improves the previous baseline results. Nonetheless, the time question has to be considered depending on the application.

Table 2: Results comparison between the method by the authors of the paper [7], of the baseline paper [4] and the proposed method.

Method [7]	**Method [4]**	**Proposed Method**
51.30%	52.15%	52.74%

Table 3: Confusion matrix of some samples tested with our method.

		Predicted		
		Negative	Neutral	Positive
Actual	Negative			
	Neutral			
	Positive			

5 Conclusions

In this work, we explore the sentiment analysis focusing on images. We achieve a result on the isolated image method that exceeds the baseline method for the same approach in the paper [4]. With this is possible to generate a sentiment classification based on the image classification. For future work we intend to further improve the method and make more tests with other datasets.

References

[1] Pretrained Models GitHub pretrained models for pytorch github. `https://github.com/cadene/pretrained-models.pytorch`. Accessed: 2019-06-17.

[2] Wyverson Bonasoli, Leyza Dorini, Rodrigo Minetto, and Thiago Silva. Sentiment analysis in outdoor images using deep learning. pages 181–188, 10 2018. doi: 10.1145/3243082.3243093.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] A. Gaspar and L.A. Alexandre. A Multimodal Approach to Image Sentiment Analysis. In *20th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, November 2019.

[5] Eduard H Hovy. Language Production, Cognition, and the Lexicon. 48:13–25, 2015. doi: 10.1007/978-3-319-08043-7. URL `http://link.springer.com/10.1007/978-3-319-08043-7`.

[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[7] Lucia Vadicamo, Carrara, and et. al. Cross-media learning for image sentiment analysis in the wild. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 308–317, Oct 2017. doi: 10.1109/ICCVW.2017.45.