SENTIMENT ANALYSIS ON FACIAL EXPRESSIONS

Chandan Uppuluri
MS in Data Science
Indiana University
chanuppu@umail.iu.edu

Rohit Dandona
MS in Data Science
Indiana University
rdandona@umail.iu.edu

Shruthi Ramakrishnan
MS in Data Science
Indiana University
shrurama@umail.iu.edu

Vignesh Sureshbabu
MS in Data Science
Indiana University
vsureshb@umail.iu.edu

Vinoth Nagabooshanam
MS in Data Science
Indiana University
vinaryan@umail.iu.edu

1. Abstract

Social media is immensely growing in today's world. For many of the social media analytics task, the sentimental analysis of online users generated content play a massive role. Earlier researchers used textual sentimental analysis to predict various analyses on measure of economic indicators, political elections etc., and now online users express their views through images and videos. Sentimental analysis of this visual content helps us to extract more user sentiments against a topic. This leads sentimental analysis from visual content is complementary to textual analysis. Two approaches to perform this analysis have been proposed. We begin by implementing a SVM classifier incorporating SIFT image descriptors and move on to implement a Convolutional Neural Network (CNN) to appropriately leverage a large scale training data set to justify the challenge of image sentimental analysis. Additionally, we test different topologies of the network to compare the change in network performance.

2. Introduction

Nowadays, social networks such as Twitter and microblog such as Weibo have become major platforms of information exchange and communication between users, between which the common information carrier is tweets. A recent study shows that images constitute about 36 percent of all the shared links on

Twitter1, which makes visual data mining an interesting and active area to explore. As an old saying has it, an image is worth a thousand words. Much alike textual content based mining approach, extensive studies have been done regarding aesthetics and emotions in images. In this paper, we are focusing on sentiment analysis to capture facial expressions from images.

So far analysis of textual information has been well developed in areas including opinion mining, human decision making, brand monitoring, stock market prediction, political voting forecasts and intelligence gathering. However, multimedia content, including images and videos, has become prevalent over all online social networks. Indeed, online social network providers are competing with each other by providing easier access to their increasingly powerful and diverse services. People with different backgrounds can easily understand the main content of an image or video. Apart from the large amount of easily available visual content, today's computational infrastructure is also much cheaper and more powerful to make the analysis of computationally intensive visual content analysis feasible.

The deep learning framework enables robust and accurate feature learning, which in turn produces the state-of- the-art performance on digit recognition (LeCun et al. 1989; Hinton, Osindero, and Teh 2006), image classification (Cires¸an et al. 2011; Krizhevsky, Sutskever, and Hinton 2012), musical signal processing (Hamel and Eck 2010) and natural language processing (Maas et al. 2011). Both the academia and industry have invested a huge amount of effort in building powerful neural networks. These works suggested that deep learning is very effective in learning robust features in a supervised or unsupervised fashion.

Inspired by the recent successes of deep learning, we are interested in solving the challenging visual sentiment analysis task using deep learning algorithms. For images related tasks, Convolutional Neural Network (CNN) are widely used due to the usage of convolutional layers. It takes into consideration the locations and neighbors of image pixels, which are important to capture useful features for visual tasks.

Convolutional Neural Networks (LeCun et al. 1998; Cires¸an et al. 2011; Krizhevsky, Sutskever, and Hinton 2012) have been proved very powerful in solving computer vision related tasks. We intend to find out whether applying CNN to image sentiment analysis provides advantages over using a predefined collection of low-level visual features or visual attributes, which have been done in prior works.

In this paper we have develop an effective deep convolutional network architecture for image sentiment analysis. We have implemented CNN using different combinations of convolution layers and fully connected layers for the prediction of image sentiment analysis.

3. Overview

As mentioned above, prediction of sentiment from visual content is complementary to textual sentiment analysis. Motivated by wide application of facial sentiment analysis in many areas such as emotion and paralinguistic communication, clinical psychology, psychiatry, neurology, pain assessment, lie detection, intelligent environments etc., here we have tried the image sentimental analysis on facial expressions. This is achieved by exploring 2 methods of classification namely Support Vector Machines and Convolution Neural Networks.

Facial Expressions

Facial expression analysis is done through computer systems. They automatically recognize facial motions and analyze facial feature changes from visual information. The following figure represents various facial expressions. It also displays facial motions and feature changes (Figure 1).

Figure 1

Data set

The dataset for the proposed model consists of 48x48 pixel grayscale images of faces. The range of each pixel of the image in the dataset varies from 0 to 255. The faces have been automatically registered. The benefit of this is that the face is more or less centered and occupies the same amount of space in each image. The training data for the model

consists of an ‘emotion’ label and image pixels. The "emotion" column consists of numeric code which ranges from 0 to 6. Similarly for “pixels” we represent it in the form of string surrounded in quotes for each image. Each string contains a space-separated pixel values in row major order. The figure shows seven categories of facial expression (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The core application of the model is to identify and categorize each face in any of the facial expression categories mentioned above based on the emotion. The training set consists of 30,000 examples, and the test set comprises of 5887 records.

4. Data Pre-Processing:

The data is a csv file. We use pandas library to extract this data. The given data is split into training set and testing set. The training set is of size 30000 and the testing set is of size 5887. This is labeled data and the labels are integers between 0 and 6. These integers must be converted to seven dimension vector to train the networks.

We have also implemented Keras, a minimalist framework designed to implement networks in an easy way. It is a python library and uses either theano or tensor flow in the backend. The use of keras (theano) gives the capability of using the gpu for the computations. The computation time is reduced when you use Theano.

5. Our initial approach (SVM classification with SIFT descriptors):

SIFT, as described in [1], consists of four major stages: **(1)** scale-space peak selection; **(2)** keypoint localization; **(3)** orientation assignment; **(4)** keypoint descriptor. In the first stage, potential interest points are identified by scanning the image over location and scale. This is implemented efficiently by constructing a Gaussian pyramid and searching for local peaks (termed keypoints) in a series of difference-of-Gaussian (DoG) images. In the second stage, candidate keypoints are localized to sub-pixel accuracy and eliminated if found to be unstable. The third identifies the dominant orientations for each keypoint based on its local image patch. The assigned orientation(s), scale and location for each keypoint enables SIFT to construct a canonical view for the keypoint that is invariant to similarity transforms. The final stage builds a local image descriptor for each keypoint, based upon the image gradients in its local neighborhood.

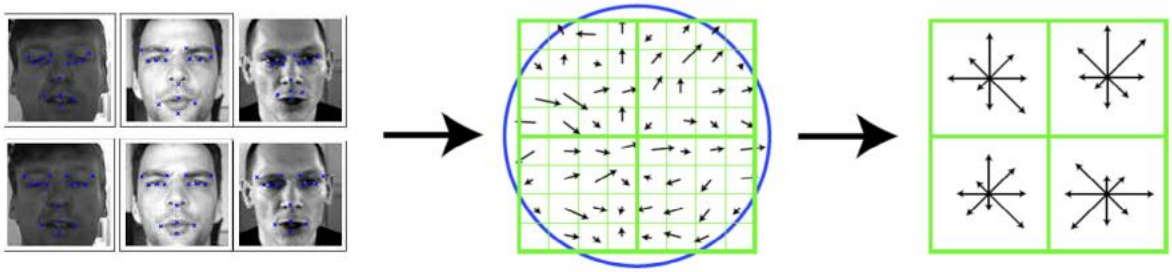

Figure 2: Image Gradients **Keypoints** **Descriptor**

k-means Clustering: After extracting features from both testing and training images, we converted vector represented patches into codewords. To do this we performed kmeans clustering over all the vectors. k-means clustering is a method to cluster or divide n observations or, in our case, features into k clusters in which each feature belongs to the

cluster of its nearest mean [2]. We cluster our features and prepare the data for histogram generation. These codewords are defined as the centers of each cluster. The number of codewords is the size of each codebook.

Histogram Generation: Each patch in an image is mapped to a certain codeword through the k-means clustering process and thus, each image can be represented by a histogram of the codewords. This is the final step before the actual classification, which is to generate histograms of the features extracted in each image [3]. These features are stacked according to which cluster they were clustered in by k-means clustering.

Support Vector Machine Classification: SVM classification [4] uses different planes in space to divide data points using planes. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories or classes are divided by a dividing plane that maximizes the margin between different classes. This is due to the fact if the separating plane has the largest distance to the nearest training data points of any class, it lowers the generalization error of the overall classifier [6]. The test points or query points are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

6. Observations (SVM):

- Extremely high computation time for our dataset
- Low accuracy (Table 1)
- Difference in facial features describing emotion, is too small for SIFT descriptors to identify for adequate classification
- SIFT generally doesn't work well with lighting changes and blurred images
- Disadvantages of using SVM classification include limitations in speed and size during both training and testing phase of the algorithm and the selection of the kernel function parameters.

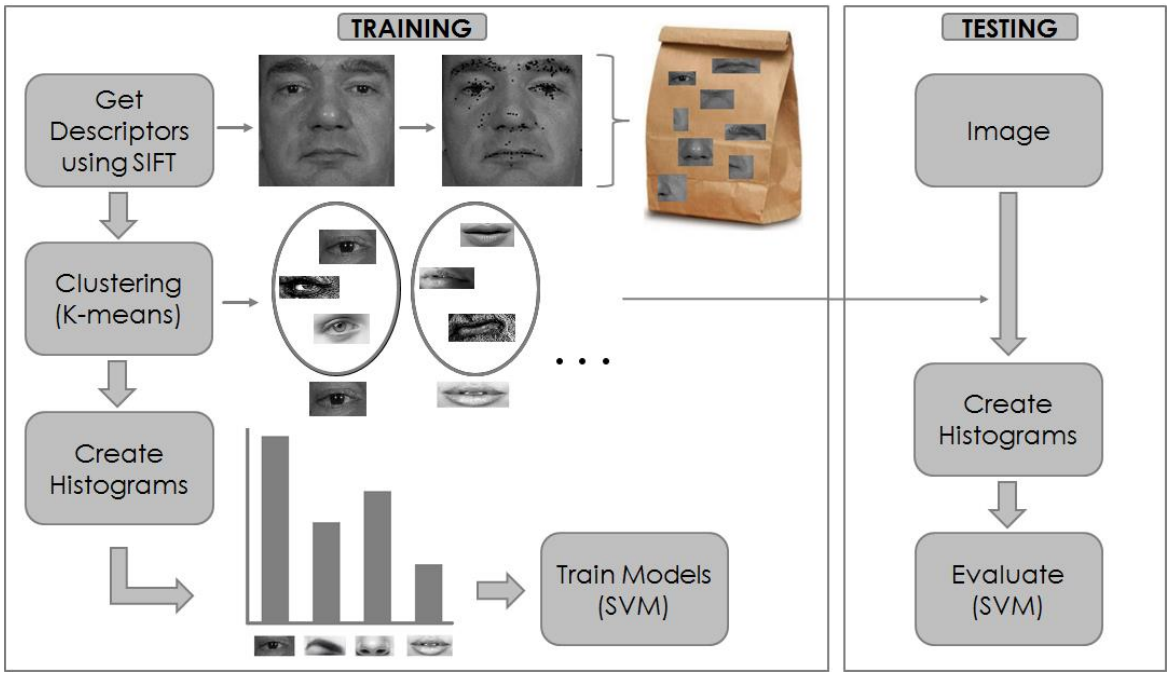

Figure 3: Training and testing via a SVM

Classifier	Accuracy % (With 50% of the training dataset)	Accuracy % (With 70% of the training dataset)
SVM	0.3845	Inconclusive (Did not finish execution)

Table 1

7. Convolutional neural networks

CNNs are hierarchical neural networks whose convolutional layers alternate with subsampling layers, reminiscent of simple and complex cells in the primary visual cortex [Wiesel and Hubel, 1959]. CNNs vary in how convolutional and subsampling layers are realized and how the nets are trained. Convolutional neural networks are often used in image recognition systems. They have achieved an error rate of 0.23 percent on the MNIST database, which as of February 2012 is the lowest achieved on the database. Another paper on using CNN for image classification reported that the learning process was "surprisingly fast" in the same paper, the best published results at the time were achieved in the MNIST database and the NORB database. When applied to facial recognition, they were able to contribute to a large decrease in error rate. In another paper, they were able to achieve a 97.6 percent recognition rate on 5,600 still images of more than 10 subjects. In 2015 a many-layered CNN demonstrated the ability to spot faces from a wide range of angles, including upside down, even when partially occluded with competitive performance. The network trained on a database of 200,000 images that included faces at various angles and orientations and a further 20 million images without faces. They used batches of 128 images over 50,000 iterations.

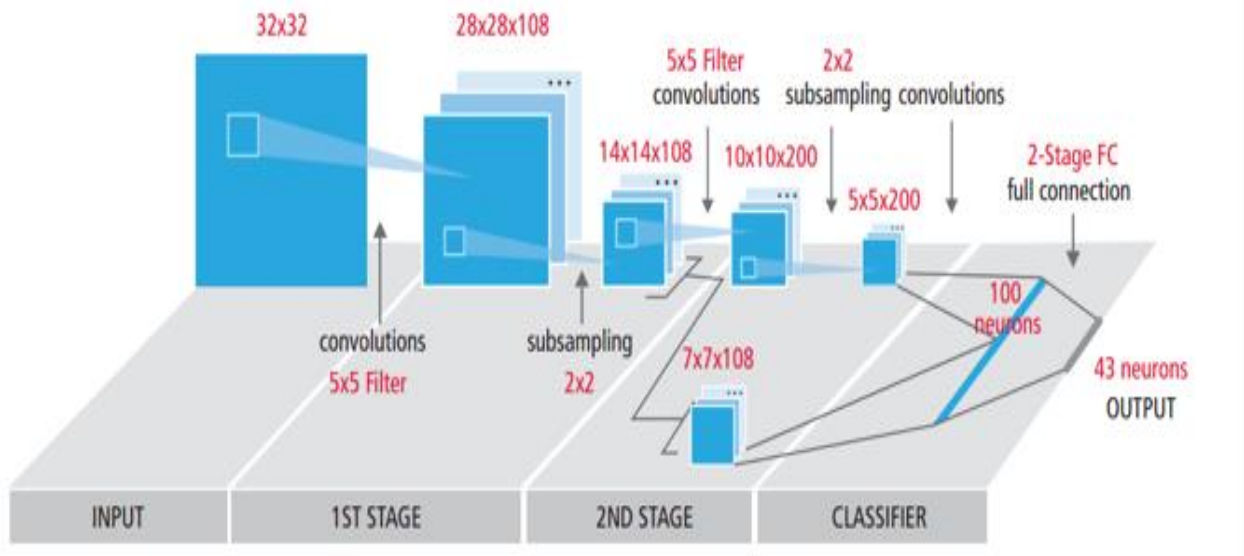

Figure 4: Typical block diagram of a CNN [4]

Our input to the convolution layer is images of size 48*48 pixels. CNN basically follows three layers.

Convolutional layer

A convolutional layer is parametrized by the size and the number of the maps, kernel sizes, skipping factors, and the connection table. Each layer has M maps of equal size (Mx, My). A kernel of size (Kx, Ky) is shifted over the valid region of the input image (i.e. the kernel has to be completely inside the image). The skipping factors Sx and Sy define how many pixels the filter/kernel skips in x and y-direction between subsequent convolutions. Each map in layer Ln is connected to at most Mn−1 maps in layer Ln−1. Neurons of a given map share their weights but have different receptive fields.

Max-pooling layer

The biggest architectural difference between our implementation and the CNN of [LeCun et al., 1998] is the use of a max-pooling layer instead of a sub-sampling layer. No such layer is used by [Simard et al., 2003] who simply skips nearby pixels prior to convolution, instead of pooling or averaging.[Scherer et al., 2010] found that max-pooling can lead to faster convergence, select superior invariant features, and improve generalization. A theoretical analysis of feature pooling in general and max-pooling in particular is given by [Boureau et al., 2010]. The output of the max-pooling layer is given by the maximum activation over non-overlapping rectangular regions of size (Kx, Ky). Max-pooling enables position invariance over larger local regions and down samples the input image by a factor of Kx and Ky along each direction.

Classification layer

Kernel sizes of convolutional filters and max-pooling rectangles as well as skipping factors are chosen such that either the output maps of the last convolutional layer are down sampled to 1 pixel per map, or a fully connected layer combines the outputs of the topmost convolutional layer into a 1D feature vector. The top layer is always fully connected, with one output unit per class label.

8. Results (CNN):

Different topological implementations have been tried with a goal of improving the accuracy.

No of convolutions	Frames	No of fully connected layers	Batch Size	Accuracy in percent
1	32: 5,5	1000, 100	1000	24.9
2	32: 5,5; 44: 5,5	1000, 300	1000	13.2
1	32: 3,3	1000, 100	1000	14
1	Increased data size for case 1			25.9

Table 2

9. Advantages of CNN

Ruggedness to shifts and distortion in the image:

Detection using CNN is rugged to distortions such as change in shape due to camera lens, different lighting conditions, different poses, presence of partial occlusions, horizontal and vertical shifts, etc. However, CNNs are shift since the same weight configuration is used across space. In theory, we also can achieve shift invariantness using fully connected layers. But the outcome of training in this case is multiple units with identical weight patterns at different locations of the input. To learn these weight configurations, a large number of training instances would be required to cover the space of possible variations.

Fewer memory requirements:

In this same hypothetical case where we use a fully connected layer to extract the features, the input image of size 32x32 and a hidden layer having 1000 features will require an order of 106 coefficients, a huge memory requirement. In the convolutional layer, the same coefficients are used across different locations in the space, so the memory requirement is drastically reduced.

Easier and better training:

Again using the standard neural network that would be equivalent to a CNN, because the number of parameters would be much higher, the training time would also increase proportionately. In a CNN, since the number of parameters is drastically reduced, training time is proportionately reduced. Also, assuming perfect training, we can design a standard neural network whose performance would be same as a CNN. But in practical training in standard neural network equivalent to CNN would have more parameters, which would lead to more noise addition during the training process. Hence, the performance of a standard neural network equivalent to a CNN will always be poorer.

10. Conclusion:

CNNs give the best performance in pattern/image recognition problems. Deep Learning also outperforms other classifiers. Based on our results, we can say that CNN gives better accuracy with One convolution layer and one hidden layer.

11. Acknowledgement:

We would like to thank Professor Muhammad Abdul-Mageed for his constant support and feedback through-out the course of the project. He has been very kind and helpful to give us valuable tips that helped us complete this paper successfully. We deeply express our appreciation to all those who have helped us in this process.

12. Références:

[1] Li, G.; Hoi, S. C.; Chang, K.; and Jain, R. 2010. Microblogging sentiment detection by collaborative online learning. In ICDM, 893–898. IEEE.

[2] Liu, B.; Dai, Y.; Li, X.; Lee,W. S.; and Yu, P. S. 2003. Building text classifiers using positive and unlabeled examples. In ICDM, 179–186. IEEE.

[3]Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. Foundations and trends in information retrieval 2(1-2):1–135.

[4]Tumasjan, A.; Sprenger, T. O.; Sandner, P. G.; and WelpeI. M. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. ICWSM 178– 185.

[5] Hu, X.; Tang, J.; Gao, H.; and Liu, H. 2013. Unsupervised sentiment analysis with emotional signals. In WWW, 607– 618. International World Wide Web Conferences Steering Committee.

[6]Siersdorfer, S.; Minack, E.; Deng, F.; and Hare, J. 2010. Analyzing and predicting sentiment of images on the social web. In ACM MM, 715–718. ACM.

[7]Borth, D.; Chen, T.; Ji, R.; and Chang, S.-F. 2013a. Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content. In ACM MM, 459–460.ACM.

[8]Yuan, J.; Mcdonough, S.; You, Q.; and Luo, J. 2013. Sentribute: image sentiment analysis from a mid-level perspective. In Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining, 10. ACM.

[9] LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324.

[10] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In NIPS, 4.

[11] LeCun, Y.; Kavukcuoglu, K.; and Farabet, C. 2010. Convolutional networks and applications in vision. In ISCAS, 253–256. IEEE

[12] CL Liu, K Nakashima, H Sako, H Fujisawa Handwritten digit recognition: benchmarking of state-of-the-art techniques

[13]Quanzeng You; Jiebo Luo; Hailin Jin ;Jianchao Yang. Robust Image Sentiment Analysis Using Progressively Trained and Domain Transferred Deep Networks.

[14] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, Yoshua Bengi. Theano: A CPU and GPU Math Compiler in Python.

13. Work Break-down:

Concept of the Project: The ground work was done by the entire team. Organized two brain storming sessions and we read quite a few research papers and decided on the final topic of the project.

Experiments and Methodology: Support Vector Machines and K means along with SIFT Descriptors was implemented by Rohit Dandona and Vignesh Sureshbabu (Including data pre-processing).

Convolution Neural Networks was implemented by Shruthi Ramakrishnan and Chandan Uppluri.

Literature Review: The related work part was completed by Vinod. He had read up to 14 research papers on the topic of our project, and came up with a lot of ideas.

Paper Preparation: The entire team contributed for the report. Every member in the team completed their respective part of the report.

Project Meetings and Peer Review: Again the whole team pitched in here. After we finalized the topic, conducted weekly status meetings, peer review during development phase of the project.

14. Authors:

ROHIT DANDONA

Rohit Dandona is a MS in Data Science student in the department of Informatics and Computing at Indiana University, Bloomington. Among others, his skillset include developing software solutions on the Hadoop framework (Hive, Java MapReduce, Pig etc.), Hadoop administration (MapR, AWS). He is interested in exploring Data Mining, Machine Learning and Computer Vision techniques.

LinkedIn Profile: https://www.linkedin.com/in/rohit-dandona-65a936ab
Github link: https://github.com/rohitdandona

CHANDAN UPPULURI

Chandan is a MS in Data Science student in the department of Informatics and Computing at Indiana University, Bloomington.

Github link: https://github.com/chandanchotu/

SHRUTHI RAMAKRISHNAN

Shruthi Ramakrishnan is a graduate student pursuing her Master of Science degree in Data Science. She is dedicate to her tasks and inquisitive to learn new things. Her skill set is varied, ranging from experience in Data Mining, web development, graphic designing, PeopleSoft ERP to Reporting tools. She was extremely involved in her undergrad outside of academics holding the position of College Arts Chief. Her hobbies include dancing, painting and anything that involves art.

LinkedIn Profile:
https://www.linkedin.com/in/shruthi-ramakrishnan-03374259

VIGNESH SURESHBABU

Vignesh Sureshbabu is a Data Science Masters student at Indiana University. Extremely organized who is focused on producing results. Realistic when setting goals, consistently develop ways to efficiently achieve, and often exceed, those goals. Area of interests are Data Mining Text Mining, and Machine Learning

Github link: https://github.com/vignesh1091

VINOTH NAGABOOSHANAM

Vinod is a MS in Data Science student in the department of Informatics and Computing at Indiana University, Bloomington.