Facial Emotion Recognition using Deep Learning

Karthik Maharajan Sankara Subramanian
Department of Computer and Information Sciences and Engineering
University of Florida
Gainesville, Florida, USA
skarthikmaharaja@ufl.edu

Abstract— ***Facial emotion recognition is one of the most important cognitive functions that our brain performs efficiently. Facial emotion is an important cue for sensing human emotion and intention in people. With the recent development in human computer interaction, an efficient facial emotion recognition system has applications in a wide variety of domains including judicial systems, interactive games, online/remote education, entertainment, and other intelligent systems. The difficulty of this problem lies in its interdisciplinary nature. Machine learning, psychology and neuroscience are involved in teaching computers to detect facial emotions. The problem involves two important tasks in pattern recognition: feature extraction and classification. Recent discoveries have made it possible to efficiently train deep neural networks with multiple layers of hidden units, allowing for more hierarchical, incrementally abstracted representations. The project uses two state-of-the-art deep learning algorithms: convolutional neural networks (CNN) and deep belief networks (DBN) to classify facial images into one of seven facial emotion categories. The convolutional neural network gives the best results consistent with recent research results in the area of image recognition.***

Keywords— ***Facial Emotion Recognition, Convolutional Neural Networks, Deep Belief Networks, Restricted Boltzmann Machine***

1. INTRODUCTION

Though speech is the primary mode of communication between humans, they also use body gestures and emotions while interacting with each other. Facial expression is the most important exhibition of human emotions. Hence facial emotions form a vital part in nonverbal communication between humans. They play an important role in interpersonal relations and convey nonverbal cues [9,10]. Facial emotion recognition systems can be a key component of human-machine interfaces; they also find applications in behavioral science and in clinical practice. Though recent research efforts have made considerable progress in the areas of face detection, feature extraction and techniques for emotion classification, development of a fully automated system accomplishing this task is challenging.

Facial expression analysis has attracted significant attention in the computer vision community during the past decade, since it lies in the intersection of many important applications, such as human computer interaction, online education, surveillance, entertainment, crowd analytics etc. The majority of existing techniques focus on classifying 7 basic (prototypical) expressions, which have been found to be universal across cultures and subgroups, namely: neutral, happy, surprised, fear, angry, sad, and disgusted. More detailed approaches follow the Facial Action Coding System (FACS), attempting either to classify which Action Units (AU) are activated or to estimate their intensity. Fewer works follow the dimensional approach, according to which facial expressions are treated as regression in the Arousal-Valence space. A very detailed and recent review can be found in [21]. In this project, deep learning techniques are used to implement an efficient facial emotion recognition system.

The dataset used in the project is the Facial Expression Recognition 2013 (FER-2013) dataset used in the challenge conducted as part of the ICML 2013 workshop on “Challenges in Representation Learning", organized by the LISA at University of Montreal. The dataset consists of 28,709 images (48x48 pixels) of faces under 7 different types of expressions and their corresponding expression category. The test set consist of 3,589 images. The task is to classify images of humans faces into one of the 7 types of expressions.

In this project, we implement two approaches based on deep learning for facial expression recognition viz. Convolutional Neural Networks (CNN) and Deep Belief Networks (DBN). The input into our system is an image; then, we use the deep learning algorithm to predict the facial expression label which should be one of these labels: anger, happiness, fear, sadness, surprise, disgust and neutral.

2. Description

2.1 Facial Emotion Recognition

Automatically perceiving and recognizing human emotions has been one of the key problems in human-computer interaction. Its associated research is inherently a multidisciplinary enterprise involving a wide variety of related fields, including computer vision, speech analysis, linguistics, cognitive psychology, robotics and learning theory, etc. [1]. A computer with more powerful emotion recognition intelligence will be able to better understand human and interact more naturally. Many real world applications such as commercial call center and affect-aware game development also benefit from such intelligence. Possible sources of input for emotion recognition include different types of signals, such as visual signals (image/video), audio, text and bio signals. For vision based emotion recognition, a number of visual cues such as human pose, action and scene context can provide useful information. Nevertheless, facial expression is arguably the most important visual cue for analyzing the underlying human emotions. Despite the continuous research efforts, accurate facial expression recognition under un-controlled environment still remains a significant challenge. Many early facial recognition datasets [2, 3, 4, 5, 6] were collected under "lab-controlled" settings where subjects were asked to artificially generate certain expressions [7]. Such deliberate behavior often results in different visual appearances, audio profiles as well as timing [8], and is therefore by no means a good representation of natural facial expressions [7]. On the other hand, recognizing facial expressions in the wild can be considerably more difficult due to the visually varying and sometimes even ambiguous nature of the problem. Other adverse factors may include poor illumination, low resolution, blur, occlusion, as well as cultural/age differences.

2.2 FER-2013 Dataset

In the ICML facial expression recognition contest [11] the challenge was to develop the best facial emotion recognition system capable of classifying images of human faces. To avoid the issue of overfitting, the contest used a completely new dataset. The main motive behind the contest was to compare feature learning methods with hand-engineered features.

This contest introduced the Facial Expression Recognition 2013 (FER-2013) dataset. FER-2013 [12] was created by Aaron Courville and Pierre Luc Carrier. It is part of a larger ongoing project. The dataset was created using the Google image search API. The search for images of faces matching a set of keywords related to various emotions were combined with factors like gender, age or ethnicity, to obtain nearly 600 strings which were used as facial image search queries. The first 1000 images obtained for each query were kept for the next stage of processing. OpenCV face recognition was used to obtain the face bounding boxes in the images. The incorrectly labeled images were rejected, duplicate images were removed and the image cropping was corrected if needed. These were then resized to 48x48 pixels and converted to grayscale. Mehdi and Goodfellow prepared a subset of the images for this challenge. They mapped the fine-grained emotion keywords into the same seven broad categories used in the Toronto Face Database [13]. The resulting dataset contained 35887 images. Of these 28,709 images constitute the training set and 3,589 images constitute the test set used in the project. The human accuracy on this dataset is determined to be 65.5%.

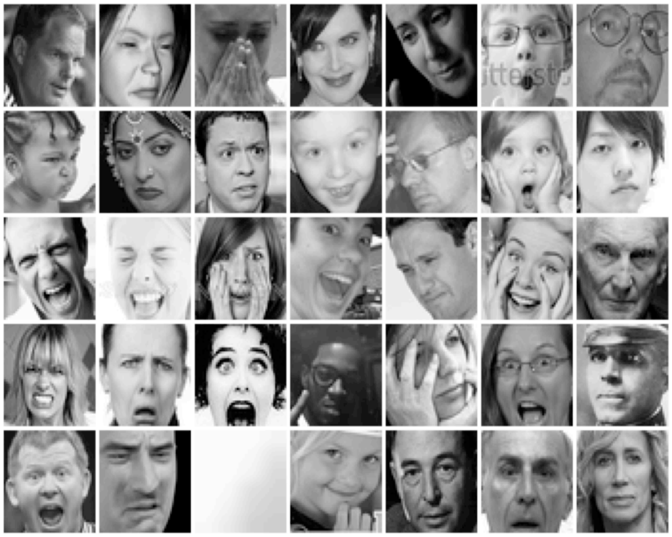


Figure 1. FER-2013 data. Each column consists of faces with the same expression: starting from the leftmost column: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral.

Label	Training set	Testing set
Angry	3995	491
Disgust	436	55
Fear	4097	528
Happy	7215	879
Sad	4830	594
Surprise	3171	416
Neutral	4965	626

Table 1: FER-2013 data

2.3 Deep Learning

Deep learning involves computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These algorithms have produced state-of-the-art performance in various areas of pattern recognition including image recognition, object recognition and speech recognition. Deep learning methods have the ability to extract intricate structures in large datasets by using the backpropagation algorithm to indicate how the internal parameters of a machine should change to compute the representation in the previous layer.

2.4 Convolutional Neural Networks

A CNN architecture (see Figure 2) is formed by stacking distinct layers that transform the input volume into an output volume using a differentiable function.

Convolutional layer

This layer is the core component of a CNN. Its parameters are a set of learnable filters or kernels. The filters have a receptive field which extends thorough the input. In the forward pass, every filter is convolved across the height and width of the input and the dot product between the filter entries are computed producing a 2-d activation map corresponding to it. Hence the CNN learns filters which activate when they see a specific type of feature in the input. Then the activation maps are stacked along the depth dimension to form the complete output of the convolution layer. An entry in the output volume can be interpreted as a neuron which works on a small input region and shares layer parameters with neurons in the same activation map.

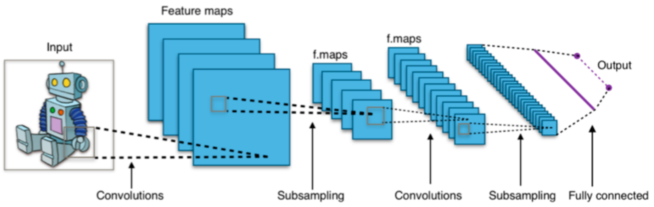
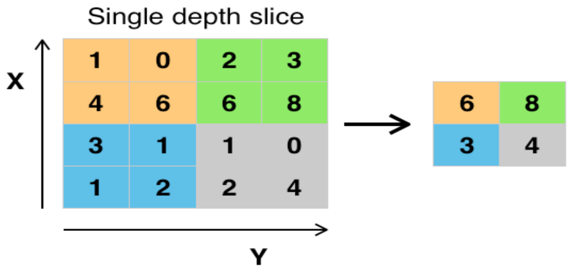


Figure 2. Architecture of a typical Convolutional Neural Network

Local Connectivity: Images are high-dimensional data and when the input to the CNN is such data, it is not practical to connect every neuron to all the neurons in the previous volume. Instead each neuron is connected only to a limited region of the input volume thus exploiting the spatial locality of the data. The extent to which this connectivity is done is a hyperparameter of the neurons and is referred to as their receptive field.

Pooling layer



1	0	2	3
4	6	6	8
3	1	1	0
1	2	2	4

6	8
3	4

Figure 3. Max pooling with a 2x2 filter and stride = 2

The pooling layer allows a number of features to be diminished and non-overlapped. It reduces spatial resolution and thus naturally decreases the importance of exactly where a feature was found, just keeping the rough location. Max-Pooling is the most commonly used function for pooling (Just as long as the feature is there, take the max, as exact position is not that critical). A 2x2 pooling would do 4:1 compression and a 3x3 would result in a 9:1 compression. Pooling smoothens the data and makes the data invariant to small translational changes. Since after the first layer, there are always multiple feature maps to connect to the next layer, it is a pre-made human decision as to which previous maps the current map receives inputs from.

ReLU (Rectified Linear Units) layer

ReLU is a neuron layer that uses the non-saturating activation function $f(x) = \max(0, x)$. It increases the nonlinearity of the decision function as well as that of the network. Though other functions like $f(x) = \tanh(x)$, $f(x) = |\tanh(x)|$, (tan hyperbolic function) and $f(x) = (1 + e^{-x})^{-1}$ (sigmoid function) increase nonlinearity, ReLU is used since it makes training faster without overfitting to the training data.

Fully Connected layer

The high-level reasoning is done using fully connected layers in a neural network. Neurons in this layer are connected to all activations in the previous layer. The activations of these neurons are computed using a matrix multiplication and then a bias offset.

Loss layer

This is the last layer of the network. It specifies the penalizing scheme for the difference between the predicted and true labels. Different loss functions are used in different tasks. In this project, softmax loss is used for predicting one among the 7 classes of emotions in the project. Sigmoid cross-entropy is used to predict the probability values of the different classes.

The hyperparameters involved in a CNN include the number of filters, filter shape and max pooling shape. In the project, weight decay is used to enforce regularization.

2.5 Deep Belief Networks

A Deep Belief Network (DBN) is one common way to implement a deep neural net [14]. DBNs incrementally create the network topology layer-by-layer, starting with the input and first hidden layers. Each layer is trained with a Restricted Boltzmann Machine (RBM) [15], which is a stochastic neural network composed of a visible layer and a hidden layer. As illustrated in figure 4, an RBM learns the probability distribution of a set of observations over the hidden units. Visible units correspond to the attributes of an observation (i.e. pixel values of an image), and hidden units model the dependencies between different aspects of the observed data, which can be interpreted as non-linear feature detection.

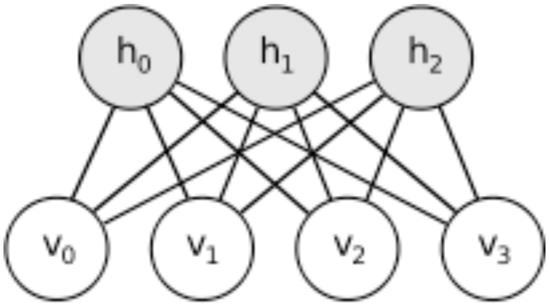


Figure 4. An illustration of the layers and connections of an RBM

DBNs are formed by stacking a series of RBMs, starting with the input layer and first hidden layer. Figure 5 shows an example of the DBN architecture. Once the first RBM is trained, its hidden layer becomes the visible layer of a second RBM and we add a new layer as the hidden layer of the new RBM. This entirely unsupervised pre-training can be continued indefinitely up to the final hidden layer. After this process is complete, connecting the final hidden layer to an output layer results in a deep neural network that can be trained with gradient descent on the error from the target output. In this way, a DBN iteratively learns increasingly abstract, hierarchical representations by composing the lower-order features of preceding layers.

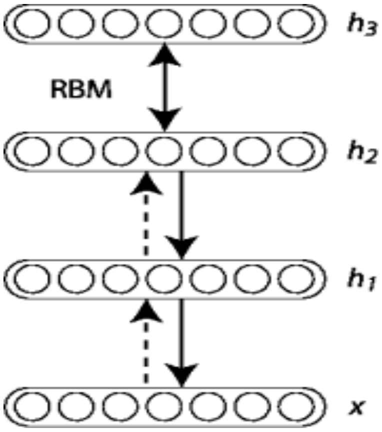


Figure 5. Architecture of a DBN

In the project, DBNs are used to perform facial emotion image classification. The levels of abstraction involved in mapping a matrix of pixel values to an emotion category make it particularly well suited for studying deep architectures. Moreover, given the compositional nature of faces (where individual features like eyes, lips, and forehead are combined in a more holistic structure), the learned intermediate representations are more qualitatively interpretable. The DBN develops low-level feature representations to compose entire faces. Once these features are learned in unsupervised training, the model efficiently learns the particular classification task. Also, this deep architecture outperforms results from shallow networks, since hierarchical representations are employed. It is clear that forcing information compression by reducing the number of units in the hidden layers results in better abstractions.

3. Evaluation

In this section, the results from the convolutional neural network algorithm are explored in greater detail since this method is currently the most successful algorithm in image classification as shown by many recent research findings. These include the performance of CNNs on the MNIST dataset [28], NORB dataset [29], and ILSVRC 2014 (ImageNet Large Scale Visual Recognition Challenge) [29]. Since the breakthrough in deep learning came from the lab of backpropagation pioneer Geoffrey Hinton's work on deep belief networks, with the discovery that pre training deep architectures layer-by-layer with unsupervised models like Restricted Boltzmann Machines (RBMs) provide a good initialization of network parameters that can then be fine-tuned with supervised data and the backpropagation training algorithm, a summary of the deep belief network implemented in the project is also provided along with the results obtained from the DBN.

3.1 Convolutional Neural Network

The convolutional neural network (CNN) is implemented in the project using TensorFlow framework built by Google. It employs a multi-layer network consisting of alternating layers of convolutions and nonlinear transformations. Then come the fully-connected layers. The final layer is the loss layer. The softmax loss is used in the project. This model achieves a peak performance of about 59.3% accuracy on the test data within a few hours of training using a CPU. The CNN network is implemented in the file fer2013.py. The training graph contains around 750 operations. The following modules are implemented,

Model inputs: Functions inputs() and distorted_inputs() read from the FER-2013 binary data files and preprocess the images for evaluation and training. We preprocess the images to 32x32 pixels before inputting them into the network. Then, they are cropped to 24x24 pixels, centrally. For tackling the insensitivity of the model to dynamic ranges these are then whitened. During training the following distortions are produced to increase the size of the dataset: random flipping from left to right, then random distortion of the brightness of the image, and then random distortion of the contrast of the image. Since the time taken to read images from disk and distorting them can slow down the training process, we run the processes as separate threads which fill a TensorFlow queue.

Model Prediction: Inference() function contains the operations which compute the logits of the prediction and primarily implements the model prediction module. It is organized as follows:

Layer Name	Description
conv 1	convolution and rectified linear activation
pool 1	max pooling
norm 1	local response normalization
conv 2	convolution and rectified linear activation
pool 2	max pooling
norm 2	local response normalization
local 3	fully connected layer with rectified linear activation
local 4	fully connected layer with rectified linear activation
softmax_linear	Linear transformation to produce logits

Table 2. Description of the layers in the network

The following is a TensorBoard graph describing the inference operation:

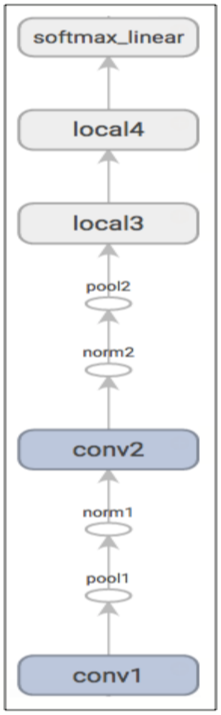


Figure 6. TensorBoard graph for inference operation

Model Training: Softmax regression is used in training. It involves applying a softmax nonlinear transformation to the network output. Then cross-entropy between the predictions and the labels are calculated. Weight decay losses are applied for regularization on all the learned variables. The objective function for the network is the sum of all the weight decay losses and the cross entropy loss. The change in total loss during training is as follows,

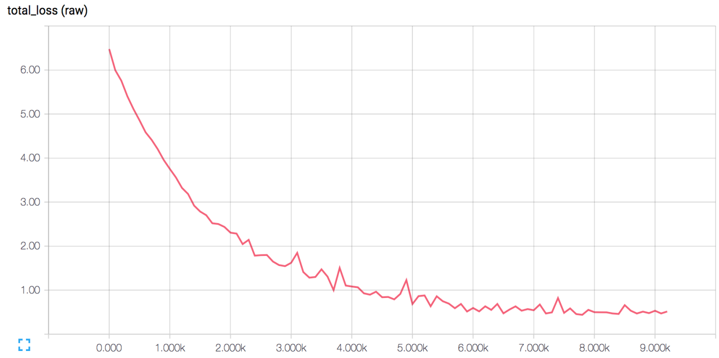
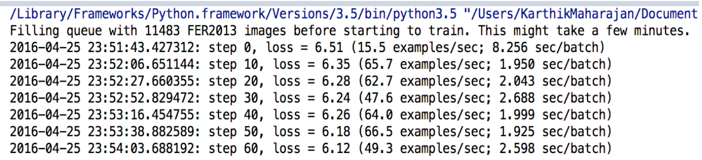


Figure 7. Total raw loss during training

The model is trained using standard gradient descent algorithm. The train() function contains the operations needed to minimize the objective by calculating the gradient and updating the learned variables. It returns an operation that executes all the calculations needed to train and update the model for one batch of images. When the training is started by running fer2013_train.py, the following output can be seen,



```
/Library/Frameworks/Python.framework/Versions/3.5/bin/python3.5 "/Users/KarthikMaharajan/Document
Filling queue with 11483 FER2013 images before starting to train. This might take a few minutes.
2016-04-25 23:51:43.427312: step 0, loss = 6.51 (15.5 examples/sec; 8.256 sec/batch)
2016-04-25 23:52:06.651144: step 10, loss = 6.35 (65.7 examples/sec; 1.950 sec/batch)
2016-04-25 23:52:27.660355: step 20, loss = 6.28 (62.7 examples/sec; 2.043 sec/batch)
2016-04-25 23:52:52.829472: step 30, loss = 6.24 (47.6 examples/sec; 2.688 sec/batch)
2016-04-25 23:53:16.454755: step 40, loss = 6.26 (64.0 examples/sec; 1.999 sec/batch)
2016-04-25 23:53:38.882589: step 50, loss = 6.18 (66.5 examples/sec; 1.925 sec/batch)
2016-04-25 23:54:03.688192: step 60, loss = 6.12 (49.3 examples/sec; 2.598 sec/batch)
```

Figure 8. Output showing network training

As can be seen from above figure, fer2013_train.py reports the total loss once in every 10 steps as well the batch processing speed. The reported loss is the average loss of the most recent batch. It is the sum of the cross entropy and all weight decay terms. fer2013_train.py periodically saves all model parameters in checkpoint files. The checkpoint file is then used by fer2013_eval.py to measure the predictive performance.

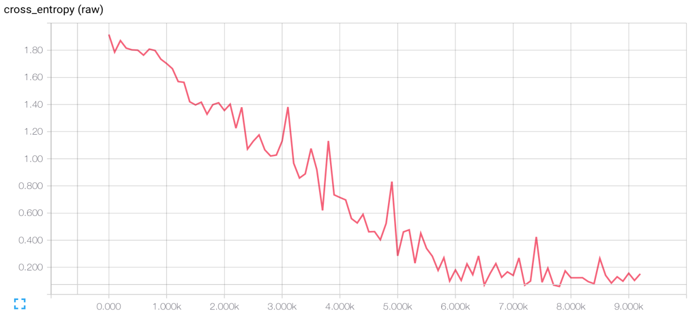
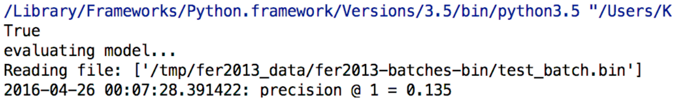


Figure 9. Total raw cross entropy during training

The model is evaluated by the script fer2013_eval.py. It uses the inference() function to construct the model. It then uses all 3,589 images in the test set of FER-2013 dataset for evaluation. It calculates the frequency at which the top prediction matches the true label of the image. It periodically reads from the checkpoint files to show the improvement of the model during training. The following output can be seen,



```
/Library/Frameworks/Python.framework/Versions/3.5/bin/python3.5 "/Users/K
True
evaluating model...
Reading file: ['/tmp/fer2013_data/fer2013-batches-bin/test_batch.bin']
2016-04-26 00:07:28.391422: precision @ 1 = 0.135
```

Figure 10. Output showing performance on test data

The following figures show the improvement in the classification accuracy of the algorithm with every step (epoch) of training.

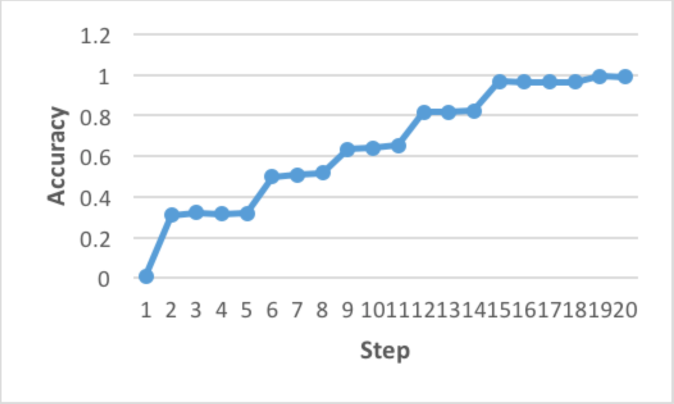


Figure 11. Stepwise change in accuracy on training set

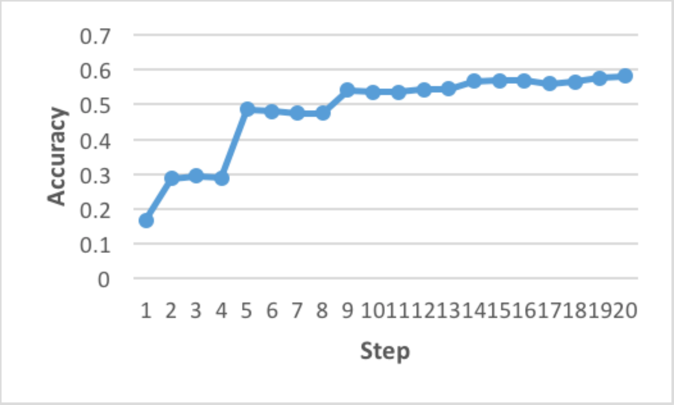


Figure 12. Stepwise change in accuracy on test set

As can be seen from the above figures, the classification accuracy of the CNN on the test set is 59.3 % and the classification accuracy of the CNN on the training set is 99.8 %.

3.2 Deep Belief Network

The deep belief network is implemented using the theano framework and other open source python libraries. When compared with other standard facial emotion recognition datasets like the JAFFE dataset [26] and the Cohn Kanade dataset [27], the FER-2013 dataset is substantially harder to classify as it is not aligned, and sometimes the images do not contain the entire face of the subject. Moreover, some of the input images do not contain a face at all or contain a distorted face. In the project, an accuracy of 49.6% on the test set is obtained, by training a deep belief net of 3 hidden layers each of size 1500 (the visible layer had an input size of 2304), with a supervised learning rate of 0.01. The accuracy on the training set is 86.3 %. The network used rectified linear units, rmsprop and Nesterov momentum (increased linearly up to 0.95). The percentage of hidden units dropped out was 50%, and the percentage of visible units dropped was out 20%. For pre-training Nesterov momentum, Gaussian visible units and Noisy rectified linear units, were used with a learning rate of 0.05, for only 1 epoch. The mini-batch size used was 20. The classification accuracy of the deep belief network is lesser than the classification accuracy of the convolutional neural network consistent with recent research findings.

	Accuracy on test data	**Accuracy on training data**
CNN	59.3%	99.8%
DBN	49.6%	86.3 %.

Table 3. Comparison of performance of the CNN and the DBN

4. Related Work

In recent years, researchers have made considerable progress in developing automatic expression classifiers [16, 17, 18]. Some expression recognition systems classify the face into a set of prototypical emotions such as happiness, sadness and anger. Others attempt to recognize the individual facial muscle movements to provide an objective description of the face. The best known psychological framework for describing nearly the entirety of facial movements is the Facial Action Coding System (FACS) [19]. FACS is a system to classify human facial movements by their appearance on the face using Action Units (AU). An AU is one of 46 atomic elements of visible facial movement or its associated deformation; an expression typically results from the accumulation of several AUs. Moreover, there have been several developments in the techniques used for facial expression recognition: Bayesian Networks, Neural Networks and the multilevel Hidden Markov Model (HMM) [20, 21]. Some of them contain drawbacks of recognition rate or timing. Usually, to achieve accurate recognition two or more techniques can be combined; then, features are extracted as needed. The success of each technique is dependent on pre-processing of the images because of illumination and feature extraction.

Deep Learning-based approaches, particularly those using CNNs, have been very successful at image-related tasks in recent years, due to their ability to extract good representations from data. Judging a person's emotion can sometimes be difficult even for humans, due to subtle differences in expressions between the more nuanced emotions (such as sadness and fear). As a result, efficient features, finely-tuned and optimized for this particular task are of great importance in order for a classifier to make good predictions. It comes as no surprise that CNNs have worked well for emotion classification, as evidenced by their use in a number of state-of-the-art algorithms for this task, as well as winning related competitions [22]. Facial expression and emotion recognition with deep learning methods were reported in [23, 24]. In particular, Tang [24] reported a deep CNN jointly learned with a linear support vector machine (SVM) output. His method achieved the first place on both training and test set data on the FER-2013 Challenge. Liu et al. [25] proposed a facial expression recognition framework with 3D CNN and deformable action parts constraints in order to jointly localize facial action parts and learn part-based representations for expression recognition

5. Summary and Conclusions

The results of the project are consistent with recent research results which show that deep learning methods particularly CNNs produce state-of-the-art performance on image recognition tasks. The accuracy of the CNN algorithm on the test data of the FER-2013 dataset is very close to the accuracy achieved by humans in the task of facial emotion recognition. In the project, the implementation is done using a dual-core CPU. In fact, the performance can be further improved by training the epoch for longer number of epochs across multiple GPUs and research results show that these implementations outperform humans. The results of the project also prove that feature learning methods which automatically learn the features from the data significantly outperform hand-engineered features for the machine learning task of classification. Also, deep learning architectures outperform shallow neural network architectures. This is evident from the following figure, which shows the relative performance of the deep learning methods implemented in the project, support vector machine with had-engineered features, and a shallow neural network.

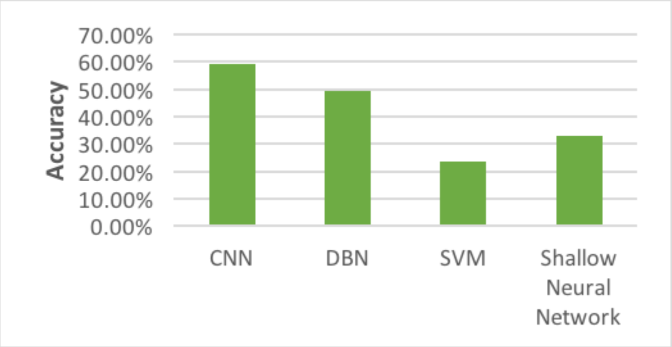


Figure 13. Relative performance of different algorithms on FER-2013 test data

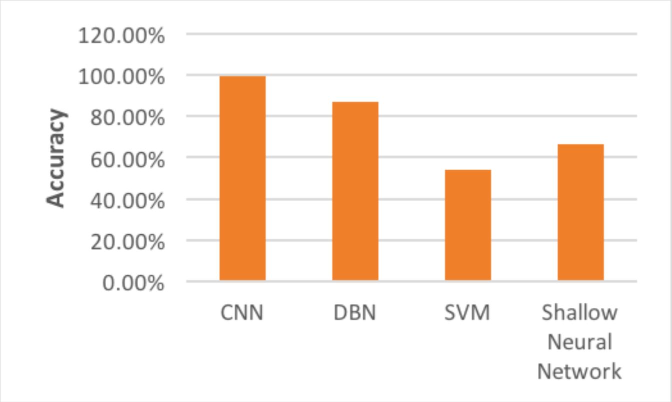


Figure 14. Relative performance of different algorithms on FER-2013 training data

Hence, the relatively new area of deep learning is exciting and poised to further grow in importance in the field of pattern recognition.

Acknowledgment

I would like to thank Dr. Dapeng Wu for providing the motivation to take up a challenging project on the important research area of deep learning and for his guidance throughout the course which helped me in the successful completion of this project.

References

[1] Z. Zeng, M. Pantic, G. Roisman, T. S. Huang, et al. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31(1):39–58, 2009.

[2] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pages 94–101. IEEE, 2010.

[3] F. Wallhoff. Facial expressions and emotion database. Technische Universita ̈t Munchen, 2006.

[4] T. Banziger and K. R. Scherer. Introducing the geneva multimodal emotion portrayal (gemep) corpus. Blueprint for affective computing: A sourcebook, pages 271–294, 2010.

[5] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan. Automatic recognition of facial actions in spontaneous expressions. Journal of multimedia, 1(6):22–35, 2006.

[6] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. Image and Vision Computing, 28(5):807–813, 2010.

[7] A. Dhall et al. Collecting large, richly annotated facial-expression databases from movies. 2012.

[8] Z. Zeng, M. Pantic, G. Roisman, T. S. Huang, et al. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31(1):39–58, 2009

[9] Bettadapura, Vinay (2012). Face expression recognition and analysis: the state of the art. arXiv preprint arXiv:1203.6722

[10] Lonare, Ashish, and Shweta V. Jain (2013). A Survey on Facial Expression Analysis for Emotion Recognition. International Journal of Advanced Research in Computer and Communication Engineering 2.12

[11] http://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge

[12] http://www-etud.iro.umontreal.ca/%7Egoodfeli/fer2013.html

[13] Joshua Susskind, Adam Anderson, and Geo_rey E. Hinton. The Toronto face dataset. Technical Report UTML TR 2010-001, U. Toronto, 2010.

[14] Geoffrey E. Hinton, Yee-Whye Teh and Simon Osindero, A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, pages 1527-1554, Volume 18, 2008.

[15] Facial Feature Tracking Under Varying Facial Expressions and Face Poses Based on Restricted Boltzmann Machines

[16] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. IEEE Transactions on Pattern

Analysis and Machine Intelligence, 23(2), 2001.

[17] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Fully automatic facial action recognition in spontaneous behavior. In Proceedings of the IEEE Conference on Automatic Facial and Gesture Recognition, 2006.

[18] M. Pantic and J.M. Rothkrantz. Facial action recognition for facial expression analysis from static face images. IEEE Transactions on Systems, Man and Cybernetics, 34(3), 2004

[19] P. Ekman,W. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologists Press, 1978

[20] Cohen, Ira, et al. ”Evaluation of expression recognition techniques.” Image and Video Retrieval. Springer Berlin Heidelberg, 2003. 184- 195.

[21] Padgett, C., Cottrell, G.: Representing face images for emotion classification. In: Conf. Advances in Neural Information Processing Systems. (1996) 894900.

[22] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. Neural Networks, 64:59 – 63, 2015. Special Issue on ‘Deep Learning of Representations’.

[23] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, . C. GÅNul.cehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari, et al. Combining modality specific deep neural networks for emotion recognition in video. In Proceedings of the 15th ACM on International conference on multimodal interaction, pages 543–550. ACM, 2013.

[24] Y. Tang. Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239, 2013.

[25] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen. Deeply learning deformable facial action parts model for dynamic expression analysis. In Computer Vision–ACCV 2014, pages 143–157. Springer, 2014.

[26] http://www.kasrl.org/jaffe.html

[27] http://www.pitt.edu/~emotion/ck-spread.htm

[28] Ciresan, Dan; Meier, Ueli; Schmidhuber, Jürgen (June 2012). "Multi-column deep neural networks for image classification". *2012 IEEE Conference on Computer Vision and Pattern Recognition* (New York, NY: Institute of Electrical and Electronics Engineers (IEEE)): 3642–3649. arXiv:1202.2745v1. doi:10.1109/CVPR.2012.6248110. ISBN 978-1-4673-1226-4. OCLC 812295155. Retrieved 2013-12-09.

[29] Ciresan, Dan; Ueli Meier; Jonathan Masci; Luca M. Gambardella; Jurgen Schmidhuber (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (PDF). *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two* **2**: 1237–1242. Retrieved 17 November 2013.

[30] "ImageNet Large Scale Visual Recognition Competition 2014 (ILSVRC2014)". Retrieved 30 January 2016.

[31] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 37(6):1113–1133, June 2015.