Computer Vision Emotion Recognition

Brenna Manning & Marissa Okoli

4.4.17

Goals

As stated in our original project proposal, our goal was to implement facial expression or emotion detection for the Neato. Our minimum goal was to achieve smile detection, but we also planned to create a classifier to detect a wider range of emotions.
We believe the neato reacting to different facial expressions and emotions expressed by a person would be an interesting and compelling experience for the person interacting with the robot.

Our original MVP was: “A basic program that could be trained as a neural network” and “a Neato that can detect smiles as well as one other facial expression and react.”

System

Our project was split into two major systems. One system identifies key-points/landmarks on the face and uses these features to recognize what emotion that face is expressing. This system uses scikit-learn for the machine learning. The other system uses a neural network to evaluate the appearance of a face and classify its expression.

This division has given us the opportunity to explore two sides of the debate of whether to use feature or appearance based methods in solving computer vision challenges around the human face. It didn’t take much research to discover that a combination of both is what actually tends to be the most effective, but we were still excited to jump into each of these methods.

For both of these emotion recognition systems we needed a significant database of faces and their corresponding emotions. We used the Cohn-Kanade (CK+) Face Database.
http://www.consortium.ri.cmu.edu/ckagree/
This database contained a large number of images of subjects expressing one of 8 emotions: neutral, anger, contempt, disgust, fear, happiness, sadness, and surprise.

While we began with a focus on identifying the difference between happiness and sadness, the goal for each of our systems was to identify which of these 8 emotion categories the facial expression from a given image best aligned with.

To make the images from the database usable for our neural net to classify, we needed to make some modifications. This was done using OpenCV. We needed the images cropped around the faces the same way for each image, convert to grayscale, and resize to a manageable size to be processed by the neural network. To crop to the face, in OpenCV we used the feature-based cascade classifier haar cascade for front facing faces. For each image we were able to create a cascade classifier using haarcascade, and then got the coordinates of the bounding box around the detected face using the built in function detectMultiScale() on this cascade and the image. From this point it was simple to crop the image to that bounding box, resize it to a constant size, and convert to grayscale. Because we used the same cascade classifier and associated functions on every image, each face was cropped the same way.

Key-point System

The key-point system was the portion of the project focused on using facial key-points to recognize an emotion. We also made the neato react based on this emotion. This system used dlib to identify key-point landmarks on a person's face for each image, generated normalized vectors based on those key-points, and used scikit-learn to make predictions based on those vectors and the labels associated with the images. How this system works is outlined more in depth in the structure section.

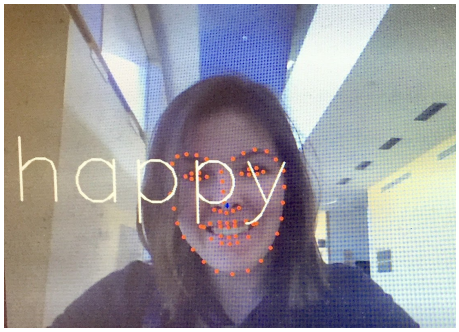
We chose to use the SVM (Support Vector Machine) machine learning classifier algorithm. We explored some alternatives to this but are happy with the SVM results.

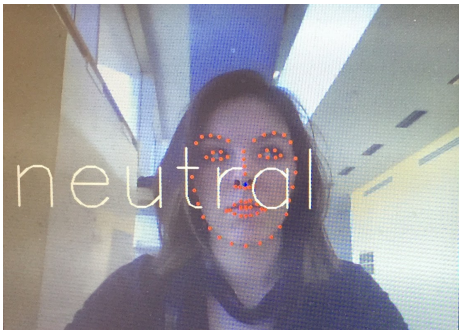
A Support Vector Machine is a memory efficient algorithm commonly used for solving classification problems. It effectively works by plotting input data as points in n-dimensional space and making classifications by separating these points using a hyper-plane. If needed, it can also plot in higher dimensions.

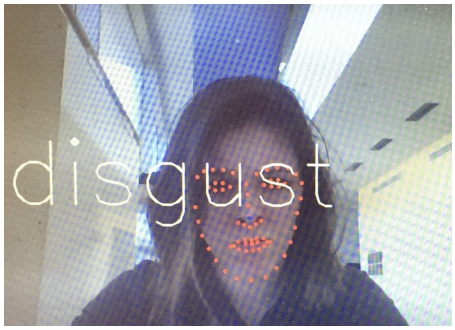
Key-point Method Results

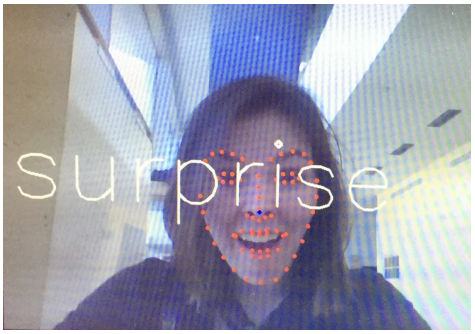
Accuracy score from train/test split for the 8 emotion categories from the CK+ dataset is about 80% using the SVM sklearn classifier algorithm.

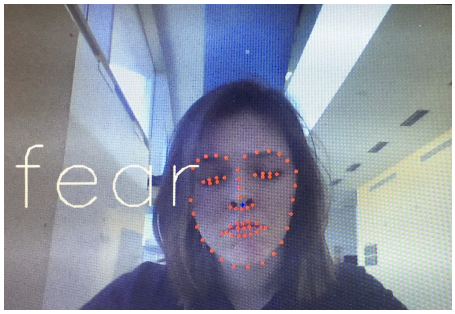
After training on the database, our classifier does a good job of identifying emotions from images of people from the webcam video capture. The screenshots below demonstrate the identification of landmark key-points and the correct classification of what emotion the face is displaying based on those landmarks calculated and shown in real time from the computer's webcam.

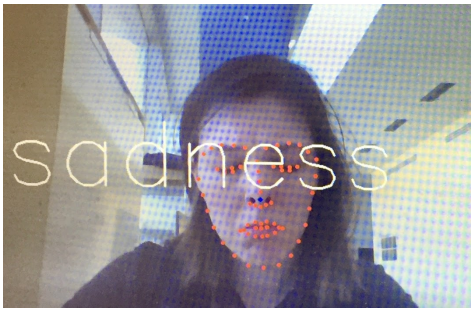










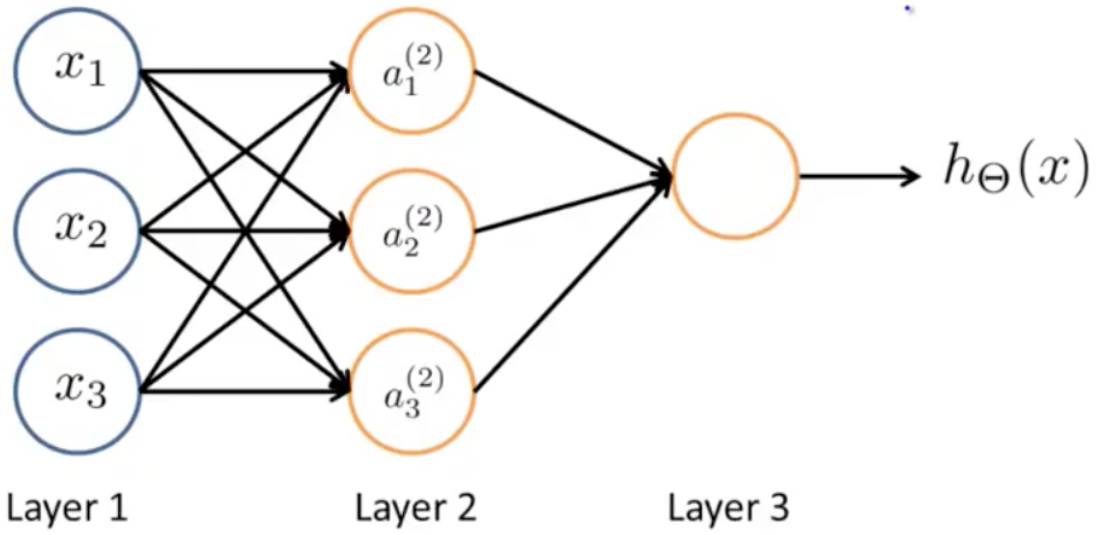


Appearance Method Neural Network

The neural network component of the project was built in Matlab where inherent matrix operations make the algorithms more computationally efficient. There are five main algorithms that comprise this system. First is the hypothesis $h_\theta(x)$. The hypothesis is the function that maps the inputs (the pixels in the image) to the output (happy, sadness, etc.). The purpose of the supervised learning algorithm is to come up with a hypothesis that fits the given data set well, but not so well that it cannot perform when presented with new inputs (the problem of overfitting a data set). The hypothesis is a function of features x_i, and the goal of the learning algorithm is to come up with parameters θ_i for each x_i that yield the most robust hypothesis. In a classification problem (i.e smile, frown, not smile, not frown) the hypothesis function is

given by $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ known as the sigmoid (logistic) activation function. To help us derive a good hypothesis function, we have the cost function $J(\theta)$ which if we minimize, will provide us with the ideal θ_i for our hypothesis. I.e. the θ_i for which $J(\theta)$ approaches zero are our ideal θ_i for our hypothesis. This is where the neural network comes in. There are two algorithms associated with the neural network: forward propagation and backward propagation. Forward propagation represents our hypothesis $h_\theta(x)$. Inputs are fed into the neural network via the input layer and propagate through its hidden layers, processed and assigned different weights (θ's) until an output is given by the output layer. In a neural network with a three-level architecture (one input layer with 3 inputs, one hidden layer with 3 units, and one output layer as shown below, the forward propagation algorithm can be calculated as follows:

Neural Network



Design Decisions

Why the two different approaches?
Early on in this project we realized that it would be more effective for both of us to divide work into sections for us to work on separately. This decision has allowed both of us to more fully meet our learning goals for the project.

What data are we using?
We debated for a while exactly what dataset we would use for this project, since that choice would influence many future choices such as whether to focus on binary classification (is the person in the photo happy or sad?) or categorical classification. We made the choice to use the CK+ dataset both because it gave us the opportunity to work on categorical classification and because the size and quality of images in this dataset worked well for key-point detection.

Key-points Design Decision
A choice had to be made about what algorithm would be used for the machine learning aspect of this problem. Early on we considered using a logistic regression. In this method we would do one-hot encoding for each label. Before jumping into this we explored classifiers provided by scikit-learn. One of these was SVM. SVM was also used successfully in blogs/tutorials we had been referencing. This seemed like a great place to start. With more time we would have liked to explore other classifiers such as K-Nearest Neighbors, etc, but we were successful with using SVM so we made the decision to concentrate on improving and expanding our project using that with the time remaining. We decided to prioritize adding interaction with the Neato over trying out different classifiers.

Structure

The following outline details the structure of the code for the key-point method of emotion recognition.

- For each image
    - Import image using OpenCV
    - Convert image to grayscale
    - Image histogram equalization using OpenCV clahe function
    - Get face and face key-points from dlib frontal face detector and using shape_predictor_68_face_landmarks.dat, a data file provided by dlib.

    - Find center of mass of key-points
    - For each key-point
        - Create vector (r, Θ) from each point to the center
          (68 vectors for each face image)
        - Alternative method of adjusting for face angle without the hard-coded values needed for accounting for head pose discovered from dlib tutorial.
          Get angle of nose bridge in the image and use this to create a correction factor.
          This means emotion detection and recognition will work for tilted faces as well as straight/upright faces.

    - Create an array for each face image of these 68 vectors
- Training: Use these arrays and the corresponding emotion labels in sklearn algorithm.
- SVM learning classifier algorithm - made to be used as classifier, did not require one-hot encoding of labels.
- Testing: get the accuracy score
- Predict: make classification predictions on new images, like from the webcam.

Challenges

Collaboration

Early on we had a difficult time meeting often and sharing work such that we were both working on something we were excited about and making useful progress. We made the choice to split this project into two separate sections so that we could each focus on the portion of the project that aligned more strongly with our learning goals. We believe this was the right decision for this project and has enabled us to be successful despite initial obstacles.

Key-point method

There were some concerns about being too dependent on existing resources, but by extending what was learned from those resources to live video and connecting this learning to an interactive robot experience, we were able to overcome that.

Neural Network method

Getting through the Coursera course on machine learning to the point where they talked about neural networks (weeks 4 and 5) was a bit of challenge given the amount of time we had. The neural network itself proved to be a highly complex algorithm with more linear algebra and differential equations than we were anticipating. Porting it over from Matlab to Python posed yet another obstacle.

Improvements and Future Work

Key-point method

With more time, we would like the chance to try out a wider variety of algorithms. It is possible another algorithm may have been even more accurate at classifying emotion recognition. SVM was pretty accurate and we are happy with its performance, but there is definitely room for improvement. It would also be interesting to spend more time creating interesting interactions between the neato and the user in terms of how the neato reacts to seeing that the user feels a certain way. This is a huge area of which we have only just scraped the surface.

Neural Network method

With more time, we would have liked to have ported the forward propagation algorithm over to Python and gotten it working on the neato.

Lessons Learned

We learned the importance of proper planning and communication. Early on we needed to figure out how best to set up this project to be a valuable learning experience for both of us given scheduling conflicts and differences in learning goals. We were happy to learn that there are ways to do this effectively. Separating the project into distinct sections was a bit uncomfortable at first, but it was the right choice for our team, and we have still learned a lot from each other throughout this process.

In projects that are driven by learning, don't be afraid to make that learning be a priority for everyone.

We also had the opportunity from this project to see the value in analyzing a problem at different levels of abstraction. The work for the key-point method had a couple large black-boxes: dlib for detection and sk-learn for learning, but it allowed for the exploration of the broader topic and integration with other work we had done in the past. On the other hand, the work for the neural network method did not allow for black boxes like that. The value was in understanding and building that algorithm, which allowed for a deeper dive into the subject.

Works Cited

Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France, 46-53.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression. Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA, 94-101.

van Gent, P. (2016). Emotion Recognition With Python, OpenCV and a Face Dataset. A tech blog about fun things with Python and embedded electronics. Retrieved from: http://www.paulvangent.com/2016/04/01/emotion-recognition-with-python-opencv-and-a-face-dataset/

van Gent, P. (2016). Emotion Recognition Using Facial Landmarks, Python, DLib and OpenCV. A tech blog about fun things with Python and embedded electronics. Retrieved from: http://www.paulvangent.com/2016/08/05/emotion-recognition-using-facial-landmarks/

Buitinck, Lars, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake Vanderplas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. "API Design for Machine Learning Software: Experiences from the Scikit-learn Project." [1309.0238] API Design for Machine Learning Software: Experiences from the Scikit-learn Project. European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases (2013), 01 Sept. 2013. Web.

Ray, Sunil, Kunal Jain, and Yogesh Kulkarni. "Understanding Support Vector Machine Algorithm from Examples (along with Code)." Analytics Vidhya. Analytics Vidhya, 13 Sept. 2016. Web.

Martinez, Aleix M. "Deciphering the Face." Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. U.S. National Library of Medicine, 2011. Web.

https://www.coursera.org/learn/machine-learning/