CS230

Gesture Evolution - Tracking Nonverbal Communication of Creative Design Teams

Kyle Yoshida
Department of Mechanical Engineering
Stanford University
kyle3@stanford.edu

Lawrence Domingo
Department of Mechanical Engineering
Stanford University
ldomingo@stanford.edu

Abstract

Communication of creative ideas is ambiguous and are often complimented with nonverbal forms of communication such as gestures or sketches. In-person team communication can take advantage of creating shared gestures and sketches more easily than in the growing remote format of hybrid work places. To understand differences in communication, we sought to track and analyze gestural communication of in-person creative design teams (CDT) in order to compare against remote CDT. Past work used Mechanical Turk to analyze video data, to our knowledge, we are the first attempt in automating the hand coding approach to analyzing video data. Using data collected on CDT, we trained a convolutional neural network to identify and flag deictic (pointing) gestures (DG). We trained our model and tested our model against two different time points separated by one month. This model could ideally be used on nascent CDTs to help them improve in their communication and overall team performance and for companies to understand how their in-person and remote CDTs work.

1 Introduction

Creative design team (CDT) work is rich with nonverbal communication (NVC) such as sketches, gestures, and prototypes. Tang (1989) found that in person mechanical engineering design work utilizes 37% gesturing to communicate creative ideas. Mabogunje & Leifer (1997) found that as ideas evolve so does verbal language. Unfortunately, CDT NVC analysis poses a problem that deals with large amounts of video data that is extremely laborious to label and analyze. Rather than manually labelling entire data sets to evaluate teams, an accurate, more efficient method is needed. Thus, to carry out this analysis, we will be using labelled videos to train a convolutional neural network (CNN) that can be used to identify when gestures occur to track gesture evolution in design teams.

This brings us to our research question: How well can we track DGs in CDT design critique meetings? We will measure the number of deictic gestures (DG) CDT expresses throughout a single design critique session. This will then enable the comparison the number of gestures used, mechanism and deictic (Karen et al., 2000), over the course of a quarter. Based on the literature, we hypothesize that CDTs that experience an decrease in deictic gesturing (DG) gesturing outperform teams that have no evolution in gestural communication. Beyond this academic study, venture capitalist or advanced projects teams can use this tool as a means to assess CDT cohesion and dynamics to further gauge the viability of a business venture. For this project, we are focusing solely on developing a CNN that can be used to identify frames in which gestures occur from video data. This can then be used in the future to output the number of DGs in a meeting.

CS230: Deep Learning, Autumn 2021, Stanford University, CA.

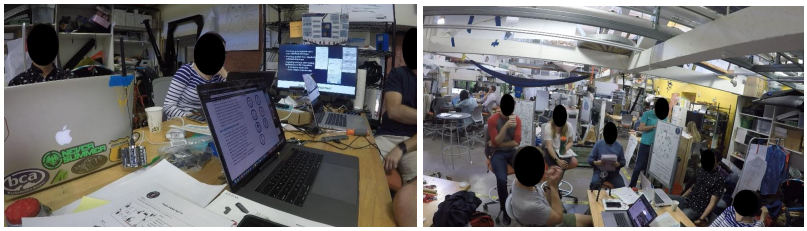

Figure 1: The left image shows the face level camera angle. The right image shows the top angle camera view. Videos were captured in situ during CDT critique meetings.

2 Related work

Assessing CDT performance has taken varying forms in the past including hand coding gestures and other non-verbal forms of communication (Tang 1989; Edelman 2011) or text-based analysis to evaluate the changes in vocabulary of CDTs over time (Mabogunje 1997; Cannon 2018). Currently, video coding of team gestures rely on hand coding each video (Tang, 1989; Edelman, 2011; Emmorey, 2000). Recent advances in deep learning make it possible to analyze unstructured data from videos to track CDT nonverbal communication. This process can be expedited through the use of cheap, large scale work services such as Mechanical Turk, however, these services can be expensive and periodically unreliable. Furthermore, larger scales of data would be too expensive to analyze and an automated approach using a neural network can help facilitate with this scale of analysis.

Gers et al. (2002) used an LSTM networks as a hidden layer of the RNN to account for temporal data in order to detect rhythmic, sequential tasks. Tsironi et al. (2017) used Convolutional Long Short-Term Memory Recurrent Neural Network (CNNLSTMRNN) in the context of gesture recognition and classified gestures according to meaning labels. However, the examples shown in the paper appeared to not be in situ. Pizzuto & Cangelosi (2019) used mask R-CNNs combined with a classical CNN to classify individual gestures rather than sequences.

Hara et al. (2018) used a 3D-CNN to model spatio-temporal data. We considered using transfer learning from this network to identify pointing in our data set. The weights from this network were trained on the Kinetics-700-2020 Human Action Dataset (DeepMind). We considered using transfer learning to tune and utilize the trained network to identify similar human actions such as pointing in our data set, but it was difficult to identify these actions in the crowded setting of our application. Narasimhaswamy et al. (2019) presented Hand-CNN which can detect hand masks and predict hand orientations in unconstrained images. Hand-CNN was able to identify hands in situ instead of in a lab setting, and the output of this CNN could be coupled with another detection algorithm.

Finally, Nair et al. (2019) implemented a simple CNN classifier consisting of 3 convolutional layers, each followed by a max pooling layer, which were fed into two fully connected layers. ReLU activation functions were used and batch normalisation is used for each layer. This CNN uses cross entropy loss as the loss function. Our work applies this network to our specific application.

3 Data Set and Features

The data set consisted of GoPro videos of small group meetings in a graduate-level design course with CDTs working on a year-long project at Stanford University. Video data (MP4) was captured during meetings over 10 weeks and were hand-labeled into two classes, times at which there was deictic gesturing (DG), and times that have no gesturing (NG). Some common gesture modes included pointing to objects, rhythmic gesturing, and modeling objects with the hands.

The data set consisted of approximately 55 hours of video data from 102 different video clips, all of which were taken from different angles for 8 CDTs Fig. 1. To prepare files for use, video data was converted from MP4 to a series of JPG images in Matlab. Furthermore, the video was discretized into 1 frame per second and downsampled to reduce file size and from 1920x1080 pixels to 64x64 pixels. This study used a 276-minute subset of the data set consisting of 14 video clips totalling 188 minutes (11,280 images) from week 5 of the course and 9 video clips totalling 88 minutes (5,280 images) from week 10 of the course. The data from week 10 of the course was split into a 70/10/20 train/dev/test distribution while the data from week 5 of the course was used wholly as test data.

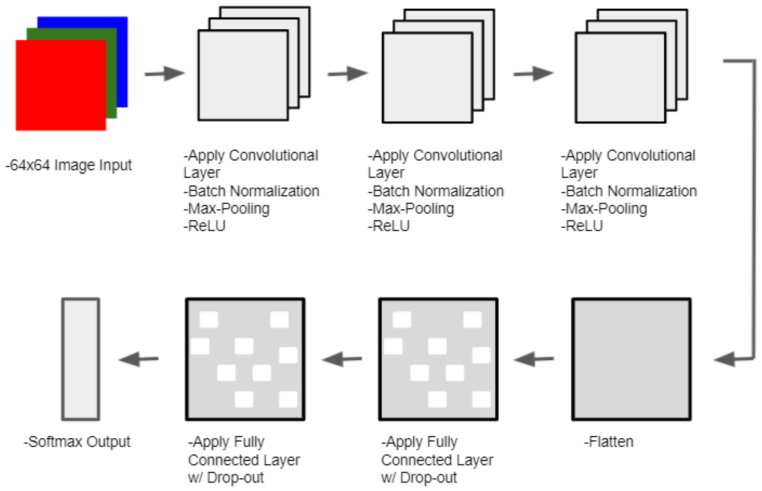

Figure 2: The architecture consisted of 3 convolutional layers followed by fully connected layers and a softmax output for binary classification.

4 Methods

A CNN classifier Fig. 2 was implemented using PyTorch consisting of 3 convolutional layers, each followed by batch normalisation, a max pooling layer, and a RelU activation function. After this, the output was flattened, then fed into two fully connected layers with drop-out. This was then fed into a softmax output containing the two classes, DG and NG. This CNN used cross entropy loss, where p is the predicted class and y is the target class:

$$L(p, y) = -(y \log(p) + (1 - y) \log(1 - p))$$

Furthermore, our data had a class imbalance with 98% of the data consisting of NG while 2% of the data consisted of DG. Due to the class imbalance, data re-sampling was used to help balance the differences between the minority and majority class. Alternative approaches include weighted cross entropy loss in which a weighting factor β can be used to mediate class imbalance. This β term could be determined via methods such as Focal Tversky loss (Abraham & Khan, 2019). However, due to the large data set size, we determined that it would be faster to re-sample the data to attain a better class balance. Thus, we re-sampled in two ways to modify the training set. The first was such that there would be twice the number of DG compared to NG samples (over-balanced) to emphasize the importance of classifying DG and increase model sensitivity, and the second was such that there would be an equal number of NG samples in comparison to the DG samples (balanced). In our case, we could have both over-sampled the minority class and under-sampled the majority class, as used in the synthetic minority over-sampling technique (Chawla et al., 2011), but we found that under-sampling of the majority class alone worked well. Thus, random re-sampling of the majority class was used to balance the data such that classes were more equal.

In the end, for training, we used data from week 7 of the 10 week long course. Thus, our "unbalanced" model used approximately 70 minutes of video data (4200 images), the "over-balanced" model used approximately 12 minutes of video data (700 images), and the "balanced" model used approximately 17 minutes of video data (1050 images). There were two test sets, one containing approximately 18 minutes of video data (1080 images) from the same distribution as the training set and one containing 188 minutes of video data (11,280 images) from a distribution not represented in the training set from week 5 of the course. The code for this work is at https://tinyurl.com/colabcodegesture (colab run files and video pre-processing file) and https://tinyurl.com/cs230colabcodegesture (.py files).

5 Experiments/Results/Discussion

To train our model, we used a learning rate of 1e-3, a batch size of 12, 10 epochs, and dropout rate of 0.8. We also tested 20 epochs, a batch size of 16, and learning rates varying from 1e-2 to 1e-4, but

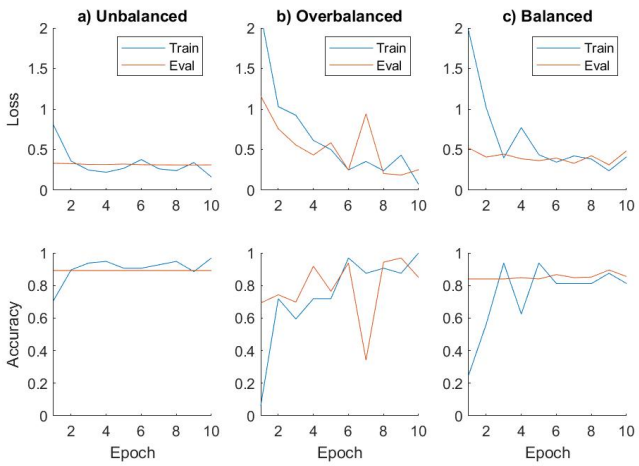

Figure 3: Accuracy and loss after each training epoch for the different weightings of training data.

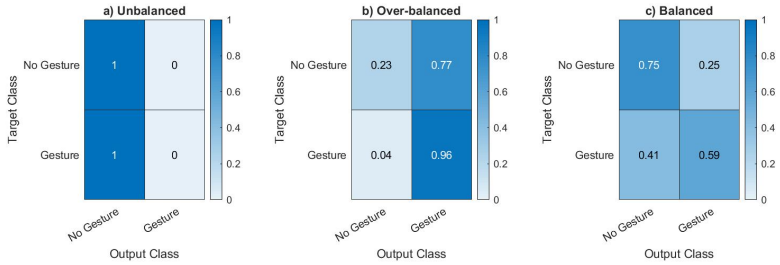

Figure 4: Different weightings of training data yielded vastly different accuracies for classification.

found that the accuracy and loss levelled off at 10 epochs and that 1e-3 yielded the best accuracy and lowest loss. We decided upon the batch size of 12, because it resulted in a model with a slightly better accuracy for the dev set (98.4% compared to 98%). This would be due to smaller batch sizes being more robust and generalizeable similar to other image analysis CNNs (Kandel & Castelli, 2020).

Our primary metric was to maximize overall accuracy (number of frames correctly classified) with a condition that at least 50% of DG and NG were identified. This metric was chosen, because of potential mislabeling of data in which DG frames were labelled based on time and a fixed 3 seconds after the time of DG occurrence. Thus, it may be possible that some occurences of DG are actually NG and some occurences of NG frames should actually be DG. Additionally, because CDT analysis still requires classification based on context of each gesture by the researcher, it would be better to have a higher sensitivity to detect DG rather than maximizing specificity. Either way, our method yielded two different training sets that the CDT researchers can use, one that helps to maximize sensitivity

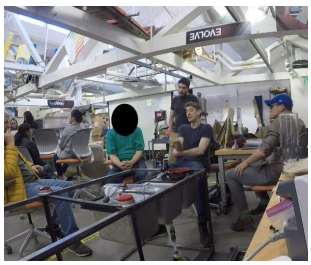

Figure 5: False negative examples occurred in a few fringe cases, such as when it was difficult to see gesturing.

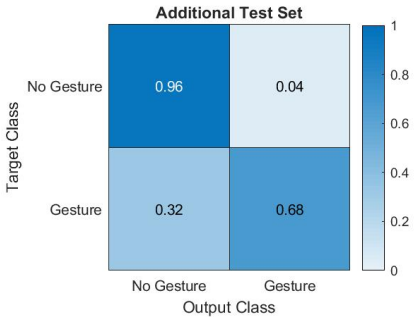

Figure 6: The balanced model was robust as it could effectively classify video data from a distribution not represented in the training set.

and another that maximizes accuracy (minimizes false positives and false negatives). Assessing CDT performance would rely on more than just NVC and using complimentary metrics such as text analysis and verbal communication analysis can help provide a more robust understanding of CDT performance.

In Figure 3, we can see the loss and accuracy over each training epoch. In the unbalanced case (Fig. 3a), we see that the training loss and accuracy improves, but due to the data imbalance, the evaluation set accuracy and loss doesn't change. In this unbalanced setup, the model simply classifies all dev set images as NG (Fig. 4a). Due to the imbalance, the loss is minimized through this biased classification method, resulting in a 98.4% overall accuracy, but it fails to recognize any images as DG. Thus, in an effort to better be able to classify DG frames, the number of DG frames were adjusted to account for double the number of NG frames in the training set. However, this resulted in overbalancing, where there is a bias towards classifying everything as DG (Fig. 4b). In this case, we correctly classify 96% of DG images as DG, but we also classify 77% of NG images as DG, resulting in an overall accuracy of 25%. Finally, in our balanced case (Fig. 4c), the model correctly classifies 75% of NG images and 59% of DG gestures, yielding an overall accuracy of 75%. When observing the changes in loss and accuracy over time, we can see that the evaluation set and the training set both show incremental improvements in loss and accuracy with the overbalanced (Fig. 3b) and balanced set (Fig. 3c), which may correspond to the fact that the model is training to learn more complex features rather than categorizing everything as NG as in the unbalanced training set. The most common reasons for misclassified images were mislabelling and fringe cases in which it was difficult to identify the gesture (Fig. 5). The extent to which the data is truly mislabelled was not conducted as this would result in evaluating thousands of images.

Finally, we conducted an additional test using data (videos from week 5) from a different distribution from the training set (videos from week 10). The results from this test (Fig. 6b) showed that the model accurately classifies 96% of NG images and 68% of DG images, yielding an overall accuracy of 94%. Thus, we can see that the model created is robust and generalizable in meeting the design requirements of the CNN.

6 Conclusion/Future Work

In this study, we found that we could create a fairly robust CNN that can identify frames from videos in which people were gesturing. We also found that to train a sensitive enough model, we need training data with balanced class representations. Another point of improvement is training the model to be sensitive enough o detect false negatives. Many DGs were missed due to blurred video frames or angle view. A more consistent camera setup can capture varying angles more consistently.

Future models trained to tag DGs should train using balanced data sets for each classification. Future work should also look to train the model on more deictic gesture data of different teams to increase the model's generalizability. Presently, the work is trained on and tested on the same teams in which there is some invariance in the data. Given the shift to hybrid and remote work, organizations should understand how non-verbal communication is altered in both in-person and remote settings when developing organizational policy. Pointing can also be tracked over time in order to study the evolution of non-verbal communication of CDTs both for in-person and remote meetings.

7 Contributions

Kyle was the primary code architect and Lawrence was secondary. Kyle wrote up the video parsing MatLab script used for pre-processing the MP4 data into JPG data. Lawrence provided the hand labelled data from a previous research project. Both team members were expected to help with pre-processing data, address administrative tasks, and writing up documents. A special thanks to Ruta for being a supportive project TA and sounding board for this project. She helped guide us in our understanding of unstructured data, computer vision, GitHub repos, and realistic deep learning project scoping.

References

[1] Tang, John C. "Listing, drawing and gesturing in design: A study of the use of shared workspaces by design teams." (1989).

[2] Mabogunje, Ade, and Larry J. Leifer. "Noun phrases as surrogates for measuring early phases of the mechanical design process." International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 80456. American Society of Mechanical Engineers, 1997.

[3] Emmorey, Karen, Barbara Tversky, and Holly A. Taylor. "Using space to describe space: Perspective in speech, sign, and gesture." Spatial cognition and computation 2.3 (2000): 157-180.

[4] Pizzuto, Gabriella ; Cangelosi, Angelo. / Exploring Deep Models for Comprehension of Deictic Gesture-Word Combinations in Cognitive Robotics. International Joint Conference on Neural Networks. 2019.

[5] Kensho Hara and Hirokatsu Kataoka and Yutaka Satoh. 2018. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://github.com/kenshohara/3D-ResNets-PyTorch

[6] Nair, S., Genthial, G., Moindrot, O., Chuang, J., and Katanforoosh, K. 2019. CS 230 - Code Examples. https://github.com/cs230-stanford/cs230-code-examples

[7] Narasimhaswamy, S., Wei, Z., Wang, Y., Zhang, J., and Hoai, M. 2019. Contextual Attention for Hand Detection in the Wild. International Conference on Computer Vision (ICCV). https://arxiv.org/pdf/1904.04882.pdf

[8] Cannon, D. M., Leifer, L. J., Cutkosky, M. R., and Ju, W., Stanford University. (2018). Prediction of design team performance through analysis of language use in meetings.

[9] D. Lee and Y. Park, "Vision-based remote control system by motion detection and open finger counting," in IEEE Transactions on Consumer Electronics, vol. 55, no. 4, pp. 2308-2313, November 2009, doi: 10.1109/TCE.2009.5373803.

[10] Johnson, Justin M., and Taghi M. Khoshgoftaar. "The effects of data sampling with deep learning and highly imbalanced big data." Information Systems Frontiers 22.5 (2020): 1113-1131.

[11] Edelman, J. A., Leifer, L. J., Banerjee, S., Beach, D. W., and Steinert, R., Stanford University. (2011). Understanding radical breaks: Media and behavior in small teams engaged in redesign scenarios.

[12] Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. doi: 10.1613/jair.953

[13] Abraham, Nabila & Khan, Naimul. (2019). A Novel Focal Tversky Loss Function With Improved Attention U-Net for Lesion Segmentation. 683-687. 10.1109/ISBI.2019.8759329.

[14] Kandel, Ibrahem & Castelli, Mauro. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT Express. 6. 10.1016/j.icte.2020.04.010.