Understanding Human Hands in Contact at Internet Scale

Dandan Shan[1], Jiaqi Geng*[1], Michelle Shu*[2], David F. Fouhey[1]
[1]University of Michigan, [2]Johns Hopkins University
{dandans,jiaqig,fouhey}@umich.edu, msh1@jhu.edu

Contents

A. Dataset

First, we describe the categories used to gather the data. These are meant to gather data and as a *likely* category of data (but are not guaranteed to be the category).

Categories: boardgames, DIY, making drinks, making food, furniture assembly, gardening, doing housework, packing, doing puzzles, repairing, and studying.

A.1. Dataset Gathering

Video Candidates: To collect the video candidates for all the videos, we generate queries by using Wordnet or frequent keywords for each category. For example, we use WordNet to search for the hyponyms of "food" and then choose verb (e.g., make, cook), location (e.g., restaurant, kitchen, home), year (e.g., 2014-2018) and sometimes some adjective to combine with it to generate our queries. However, for some categories, like repairing, taking hyponyms from WordNet does not work. Instead, we browse YouTube pages about repairing and gather frequent words for it. Finally, we generate 1,200 queries for each category and search them through the YouTube API to get responses.

Feature Representation: We find that the 4 thumbnails of each video are good representations and summarization of its content. Thus, we extract features from Alexnet `pool5` for each of the 4 thumbnails. Then, we define the video feature as the average of the 4 `pool5` feature plus the mean, min, max of distances between them.

Hand-Score Filter: The definition of Hand-Score is the percentage of the equally-spaced 100 frames from the dataset containing hands. We use Faster-RCNN as the hand detector. Then, we prepare 1,000 samples (x, y) for each category, where the x is the merged `pool5` features from Alexnet and y is the Hand-Score from Faster-RCNN to train a linear-SVR mapping x to y. This gives predicted Hand-Score for all the videos. Finally, we rank all the videos by their Hand-Score.

Interaction-Score Filter: The focus of the dataset is hand-object interaction. We not only make sure that there are hands in the videos but also we care about whether the hands are in contact with objects. We choose 12 equal-spaced frames from 1,000 video samples and concatenate them into 3 2x2 images, and manually label interaction state via the crowd (i.e., whether any of the images in the grid has interaction). The definition of Interaction-Score is the fraction of the 3 concatenated images containing in-contact hands. Then, like the Hand-score approach, we use 1,000 video samples for each category. The feature x comes from Alexnet and manually labeled Interaction-Score to train a linear-SVR to do Interaction-Score regression for all the videos. Finally, we rank all the videos by their Interaction-Score.

Cartoon Filter: After doing this, we found a lot of cartoon videos mixed in our dataset with high rankings in Hand-Score and Interaction-Score. So it is necessary to build a cartoon classifier to filter out these cartoons. We prepare a small 1000 cartoon samples (each sample is a 2x2 concatenated thumbnail image of one video) and 1000 normal thumbnail samples to train a pre-trained ResNet-50 with trainable parameters only at the last 3 layers. We use Adam optimizer with a learning rate of 1e-3, batch size 64.

Dataset Selection: We only keep the overlap between the top 20% videos from Hand-Score ranking list and the top 20% from the Interaction-Score ranking list. Afterward, we run the cartoon classifier on all the remaining videos to get

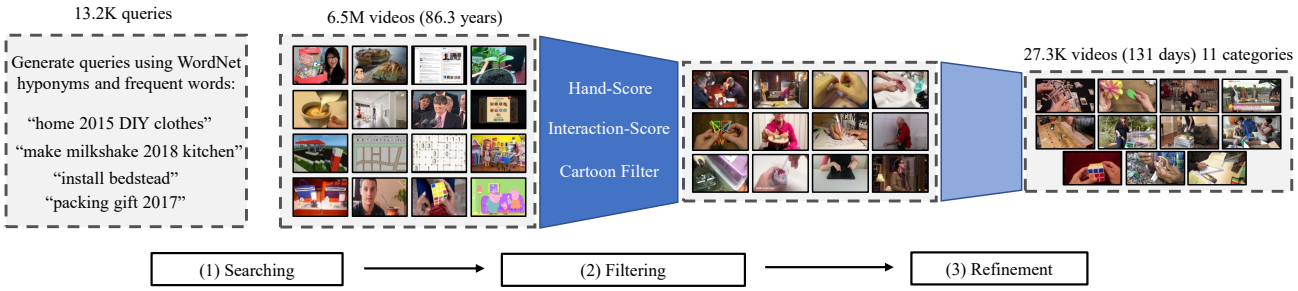


Figure 1. Dataset collection pipeline.

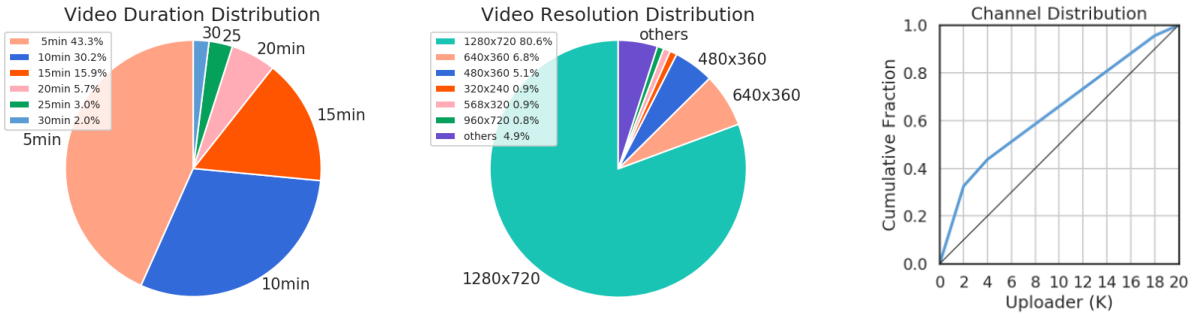


Figure 2. Dataset statistics.

Table 1. 100DOH dataset details

Category	Amount	Size
boardgame	2,654	179G
DIY	2,902	198G
drinks	2,739	155G
food	2,737	203G
furniture	2,813	145G
gardening	955	79G
housework	2,809	324G
packing	2,809	234G
puzzle	2,825	176G
repairing	2,764	177G
study	1,299	106G

rid of cartoons.

In order to make our dataset balanced, for each category, we only download up to 3,000 videos if they exist. To ensure the diversity of our dataset, we also limit that each YouTube channel provides at most 20 videos to the dataset and the maximum duration of each video may not exceed 30 minutes.

A.2. Data Annotation

Definitions of hand contact state: We define the following 5 categories for hand contact state:
(1) No contact(**N**): the hand is not contacting anything.
(2) Self contact(**S**): the person's hand is contacting his/her body (including hair).
(3) Other person(**O**): the hand is contacting another person.
(4) Portable Object(**P**): the hand is contacting an object that can be easily carried with one hand.
(5) Non-portable Object(**F**): the hand is contacting an object that cannot be easily carried with one hand, especially **furniture**.

We used a crowdsourcing platform (thehive.ai) to label our data. The process of our annotation is as follows: (1) annotate whether there are hands in the sample frames and then only keep frames containing hands (2) annotate hand bboxes in the frames and whether they are left/right; (3) annotate hand contact state; (4) for each in-contact hand, annotate the object or person that the hand is in contact with. For our 100K annotated hand-contact state video frame dataset, we make sure that each sample frame has conclusive annotations from human workers. Throughout, the platform uses qualification tasks (i.e., a test that the workers take to show they understand the direction) as well as sentinels (i.e., data with known labels that are mixed in with unlabeled data to check worker quality).

B. Model Details

We will release all models. All of our models are implemented in PyTorch.

B.1. Hand Detection Model

We used a standard Faster-RCNN as our hand detection model with pre-trained ResNet-101 as the backbone. We trained it for 8 epochs using an SGD optimizer with a learning rate of 1e-3, the momentum of 0.9 and mini-batch size of 1.

B.2. Hand and Object Detection Model

We built on Faster-RCNN by adding auxiliary layers and losses per bounding box. We include several loss terms so that we need to handle wide variance in the loss scale: (1) we normalize the magnitude by dividing it by 1000 to make it naturally a more sensible scale; (2) we multiply the orientation component by 0.1 after after normalization; (3) we scale each of L_{side}, L_{state} and L_{offset} by a factor of 0.1.

B.3. Good-Bad Classifier Model

We use a multilayer perceptron with two hidden layers of 100 and 50 each. We train it using Adam optimizer with a learning rate of 1e-4 and a mini-batch size of 8.

B.4. Hand Prediction Model

Our Hand Prediction system is as follows:

Pose Regression Model: ResNet-18 (512) $\rightarrow$ (256) $\rightarrow$ R $\rightarrow$ pose regression (33) / side regression (2). This model is supervised on pose, side and vertices (mapping pose to vertices using MANO) with weight of 10, 100, and 1 correspondingly. We train it using Adam optimizer with a learning rate of 1e-4 and batch size of 200.

Pose Classification Model: ResNet-18 (512) $\rightarrow$ Dropout(0.75) $\rightarrow$ (10). This model is supervised on class label using a cross-entropy loss. We train it using Adam optimizer with a learning rate of 1e-4 and batch size 200.

We use the global orientation (the first 3 parameter from pose regression), hand side and pose classification result together as our final prediction.

C. Experiment Details and More Results

C.1. Instructions for correct/incorrect label

We used the following definitions for annotation of whether the hand is correct or not (S5.3)

Correct Hands *have: (1) all of the fingers correctly reconstructed and (2) look very similar to the input image from the same viewpoint. The hand can still be correct even if you cannot see all the fingers. If it looks plausible, then this is also ok.*

Incorrect Hands: *(1) do not have all of the fingers correctly reconstructed or (2) do not look similar to the input image or (3) do not look like real hands.*

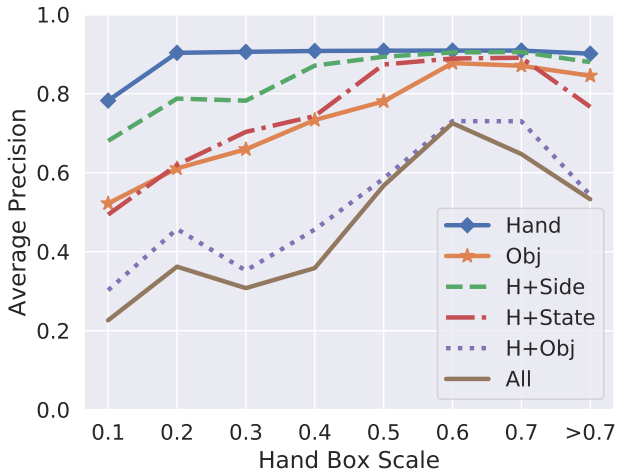


Figure 3. Analysis as a function of hand scale.

C.2. Full Hand State

We show some more examples of random full state predictions below on our dataset in Figure 5 and other datasets in Figure 4. Hand boxes tend to be accurate. Current issues with the system tend to be getting the hand contact state correct and correctly associating boxes with hands when there are lots of people.

We also show the full hand state performance as a function of hand scale in Figure 3. We divide images into different hand scale bins according to the average hand scale function $\sqrt{(x \times y)/(w \times h)}$, where (x, y) is average hand size and (w, h) is image size.

C.3. Mesh Quality Assessment

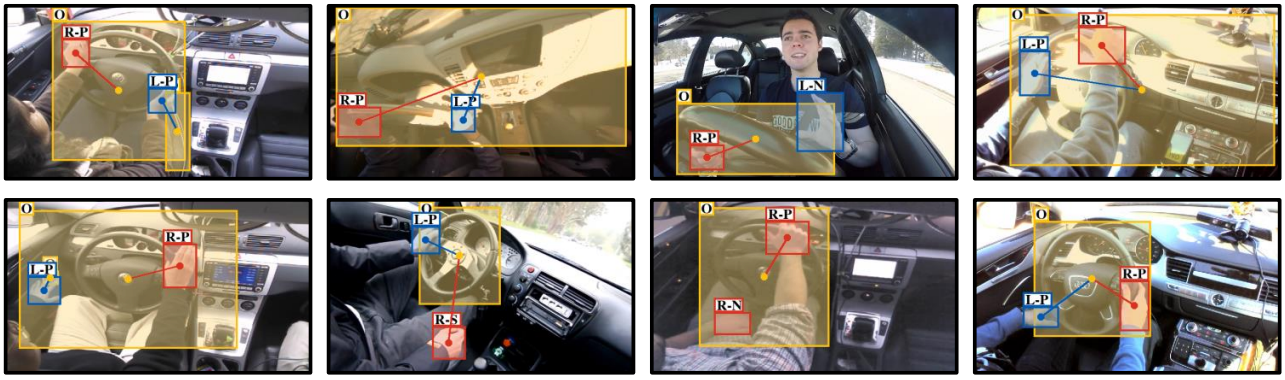
We show some additional examples of good/bad classification of meshes in Figure 6.

C.4. Hand Mesh Prediction

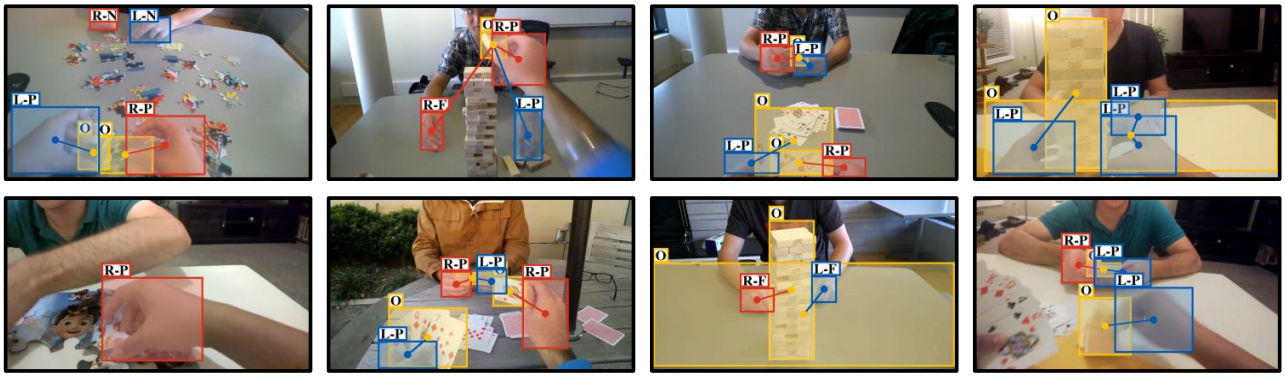
We show some additional examples of predictions of hands from objects in Figure 7 as well as the centroids we use in Figure 8.

Figure 4. Random full hand state results.

VIVA



Ego



VGG

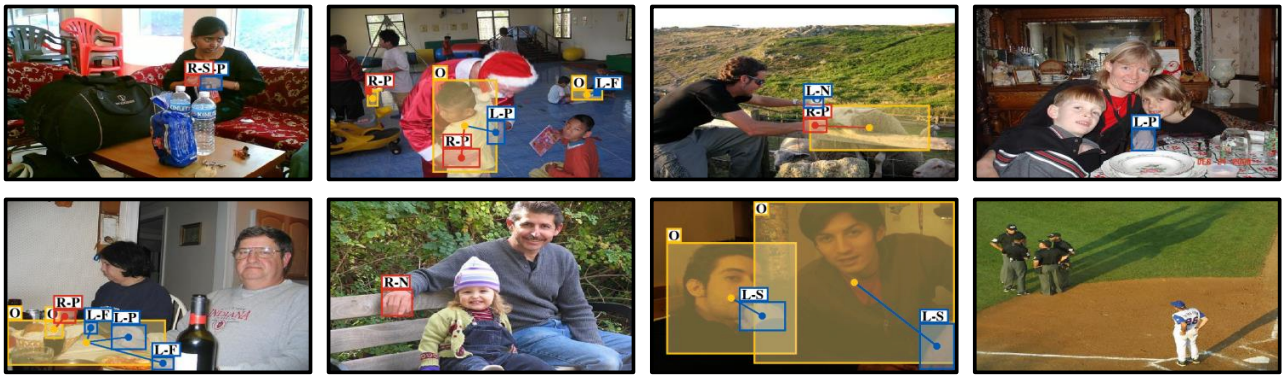


Figure 5. Random full hand state results on other datasets.

Figure 6. Random mesh quality assessment results.

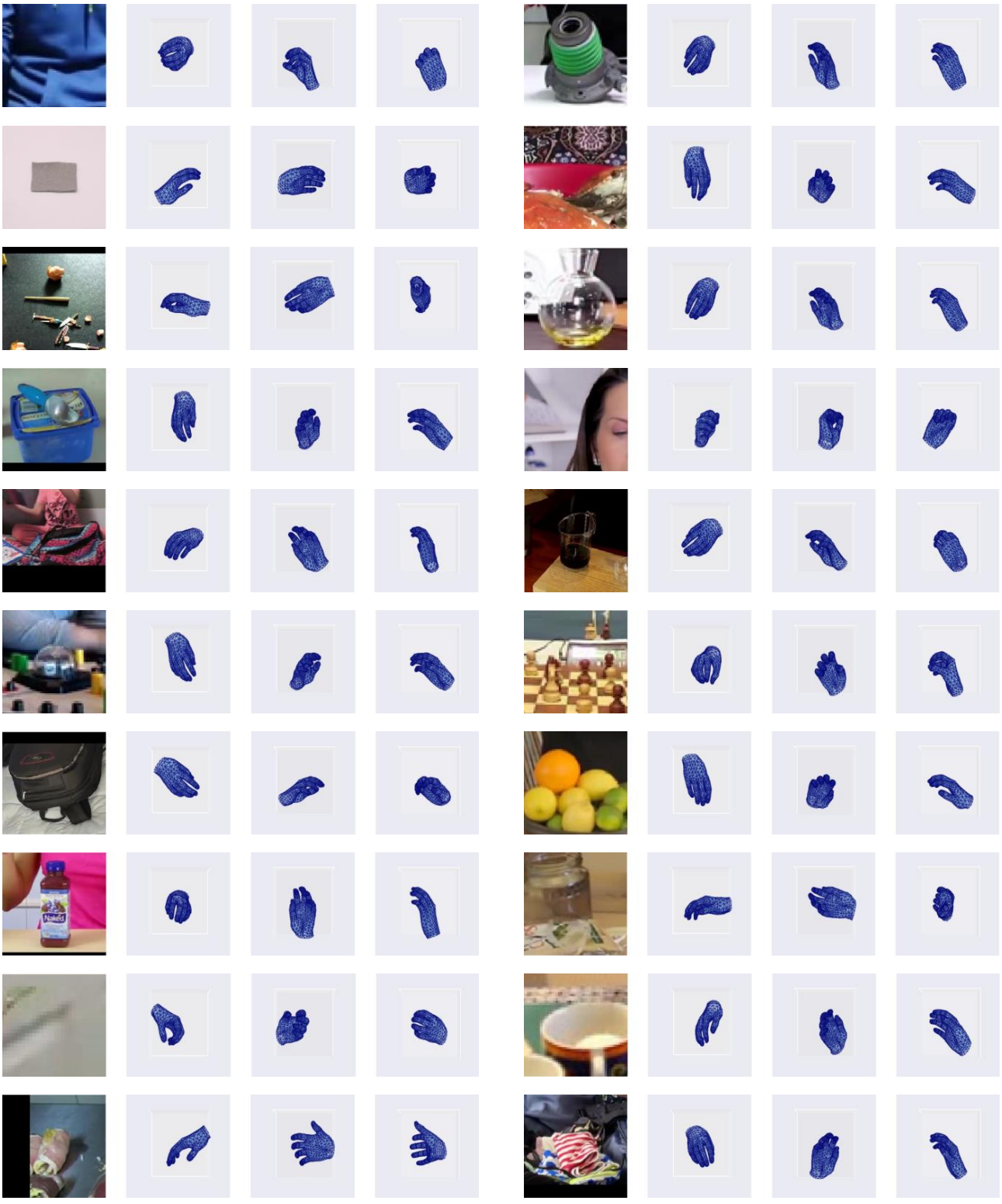


Figure 7. Random mesh prediction results.

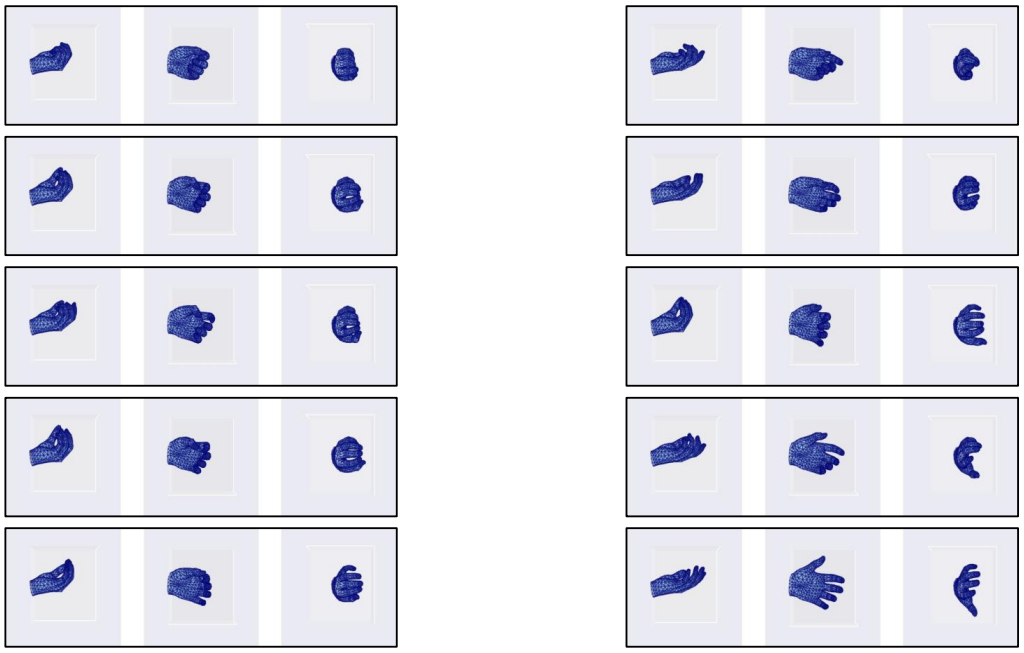


Figure 8. 10 centroids for classification.