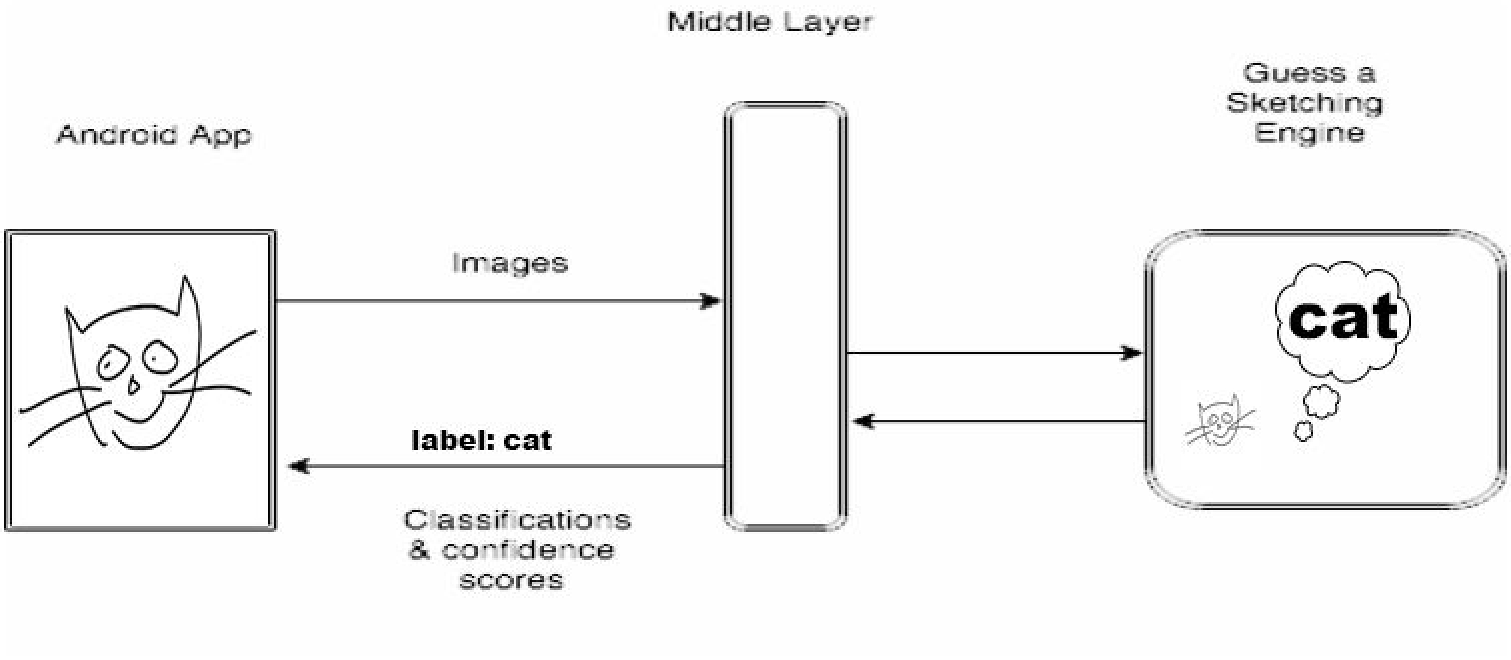
Guess A Sketch

Applying Machine Learning To Predict Sketched Objects

Introduction

- System designed for predicting objects which the user is trying to sketch
- System gives feedback of what it thinks the object is
- The system updates its prediction as user keeps modifying the sketch
- User is able to give positive feedback if the guessing is correct. If not, they can tell the system what object they are trying to draw
- The feedback mechanism increases confidence of the system (positive feedback) or learn from it (negative feedback)

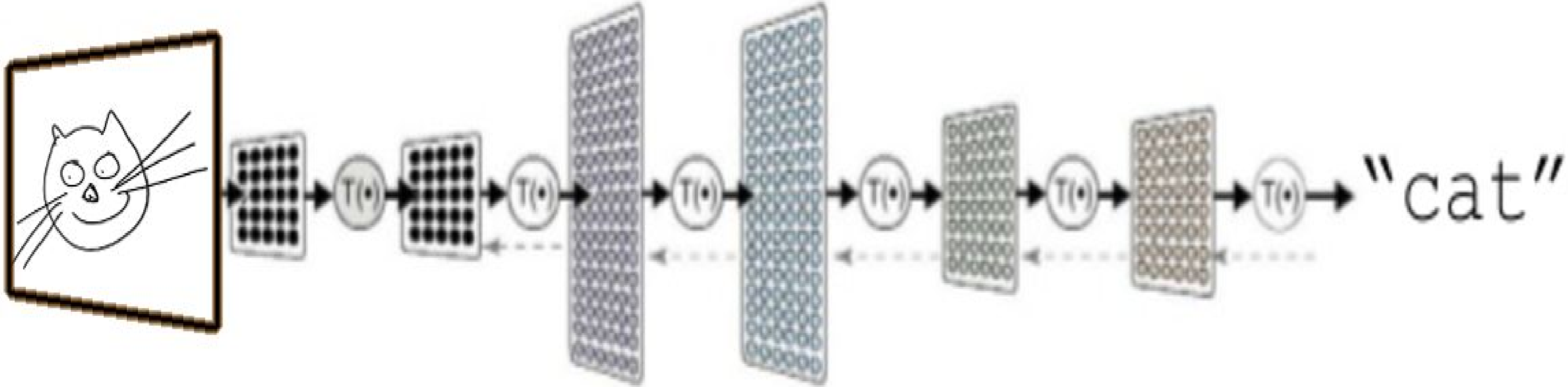
Component Diagram



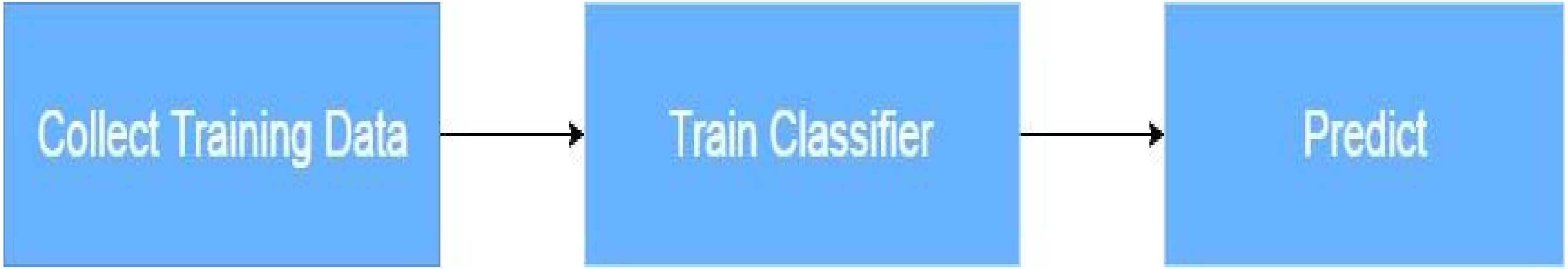
TensorFlow

- TensorFlow is an open source software library for ML which use for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays. TensorFlow is developed for the purpose of training neural networks to detect patterns and correlations, in which the Computation code is written in python, java or c++.
- Deep convolutional neural network
- Once Trained, it gives excellent predictions

Deep Convolutional Neural Network



Classifier Training



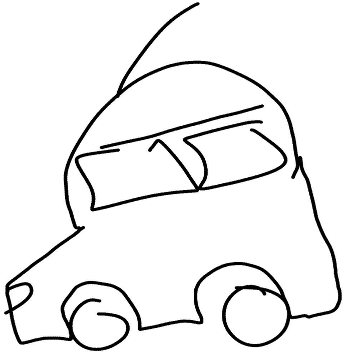
Collect Training Data

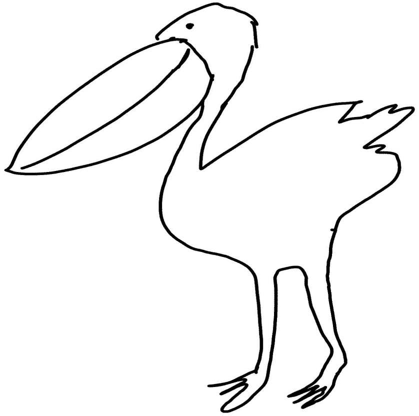
Train Classifier

Predict

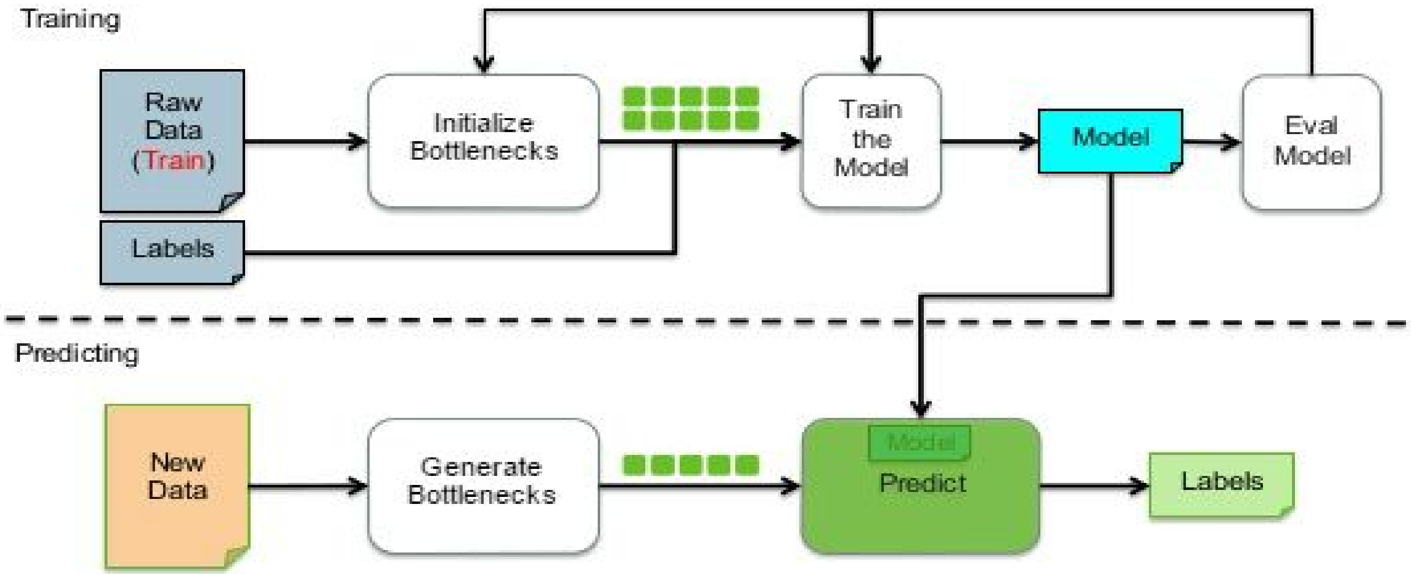
Training Data

- Created an archive of common images into 12 categories of objects





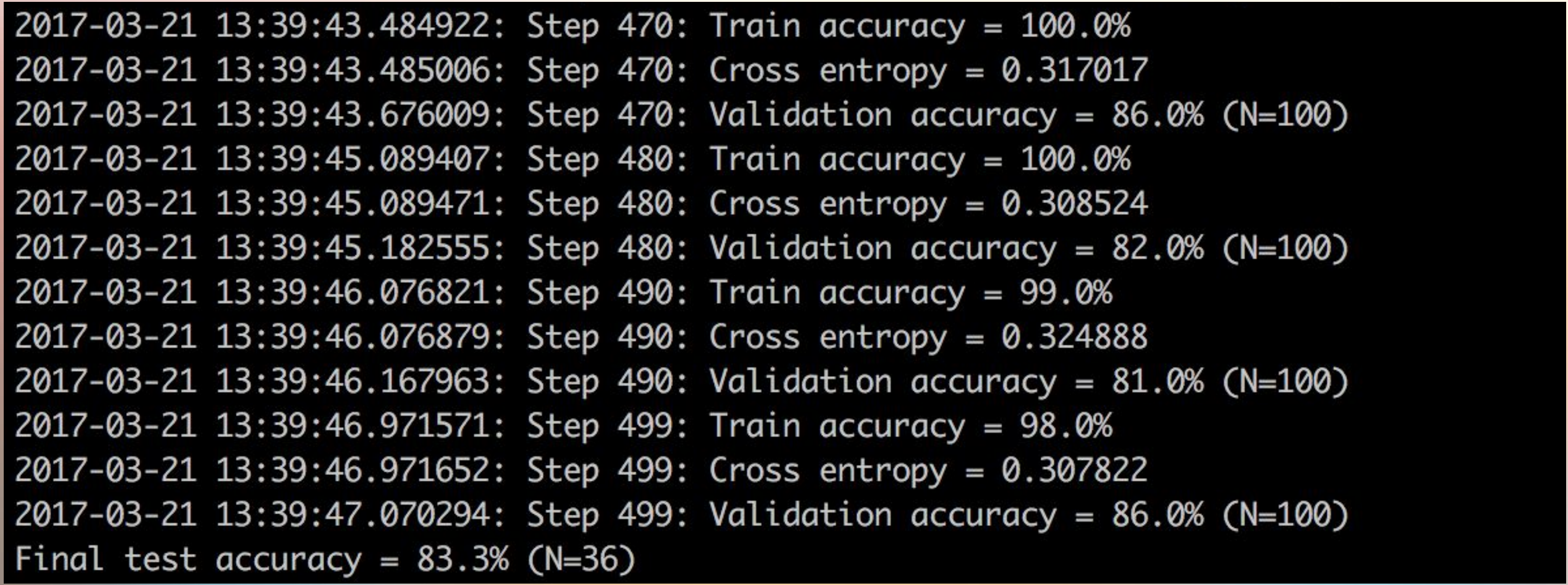
Supervised Learning Workflow



Training The System

- The training phase analyze all the images and creates bottleneck files for each image
- Took 25 minutes to train the system with 376 images
- Once trained, we test the classifier with the initial testing data set
- Training accuracy: shows the percentage of training batch that were labeled with the categories
- 83.3% training accuracy
- Validation accuracy: shows the precision of correctly labeled image

Training Accuracy



```
2017-03-21 13:39:43.484922: Step 470: Train accuracy = 100.0%
2017-03-21 13:39:43.485006: Step 470: Cross entropy = 0.317017
2017-03-21 13:39:43.676009: Step 470: Validation accuracy = 86.0% (N=100)
2017-03-21 13:39:45.089407: Step 480: Train accuracy = 100.0%
2017-03-21 13:39:45.089471: Step 480: Cross entropy = 0.308524
2017-03-21 13:39:45.182555: Step 480: Validation accuracy = 82.0% (N=100)
2017-03-21 13:39:46.076821: Step 490: Train accuracy = 99.0%
2017-03-21 13:39:46.076879: Step 490: Cross entropy = 0.324888
2017-03-21 13:39:46.167963: Step 490: Validation accuracy = 81.0% (N=100)
2017-03-21 13:39:46.971571: Step 499: Train accuracy = 98.0%
2017-03-21 13:39:46.971652: Step 499: Cross entropy = 0.307822
2017-03-21 13:39:47.070294: Step 499: Validation accuracy = 86.0% (N=100)
Final test accuracy = 83.3% (N=36)
```

Feedbacks

- Positive feedback is when system guess the image correctly
- It can be used in the future to improve the system performance

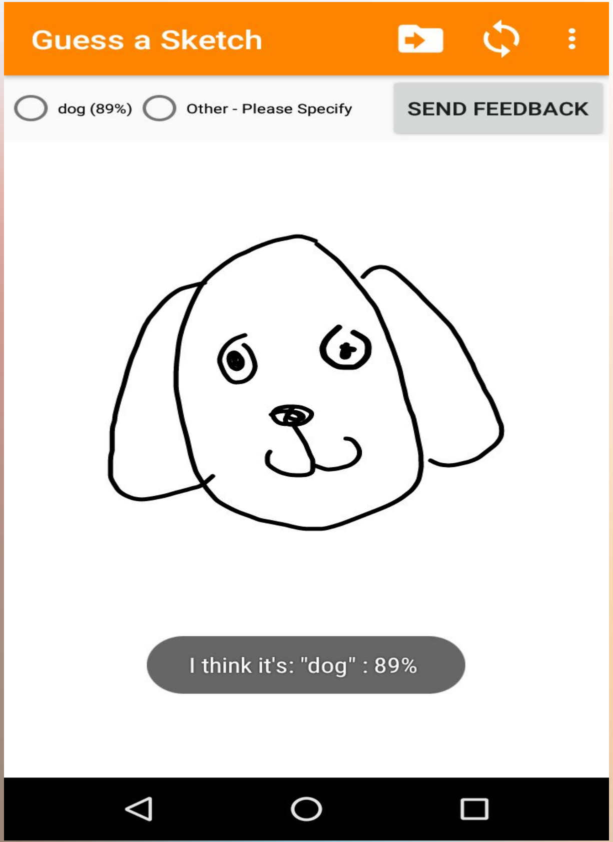
Feedbacks

- Negative feedback is when system fails to correctly guess the object sketched by user
- The system asks the user to label this image, and send his feedback along with the image to server.
- It will be used to retrain the guess sketch engine
- Based on this feedback either new label is created or sketch is added to the existing label

Re-training The System

- The system is retrained to predict the sketches of objects that are newly introduced - mostly based on user feedback
- While retraining the system, bottleneck files will be created only for the images used for retraining resulting in a very quick retrain operation
- Service is guaranteed to stay up even while retraining is taking place

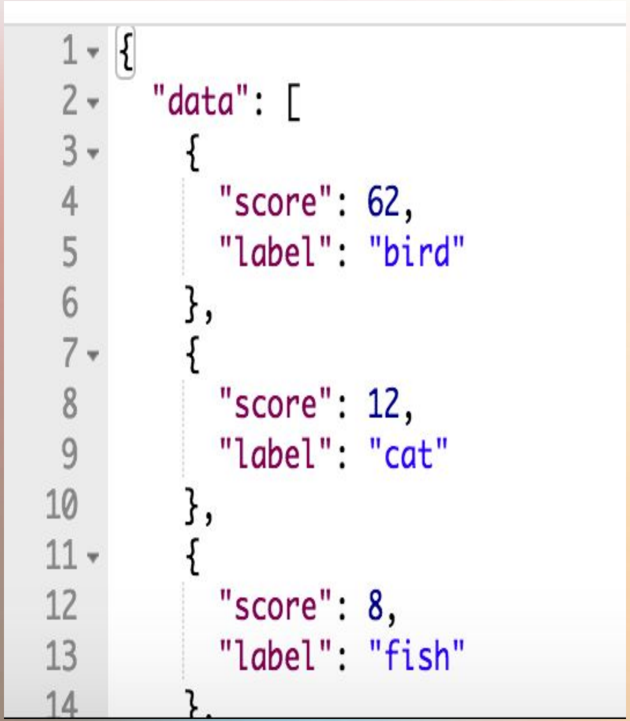
Android App



I think it's: "dog" : 89%

- It provides a canvas for the users to draw sketches on
- Sends sketch drawn by users to the middle layer
- Receive prediction of objects the sketch most likely matches to from middle layer
- Send user positive/negative feedbacks to the middle layer which is used to retrain the system

Middle Layer



```
{
  "data": [
    {
      "score": 62,
      "label": "bird"
    },
    {
      "score": 12,
      "label": "cat"
    },
    {
      "score": 8,
      "label": "fish"
    },
```

- Receives images from Android app and sends it to guess sketch engine
- Receives prediction from guess sketch engine and sends it to Android app
- Receives user feedback from Android app
- Saves the sketches drawn by users locally which is used to retrain the guess sketch engine

Guess Sketch Engine

- Uses TensorFlow python API
- Initially the engine is untrained
- Based on training it predicts objects that closely matches sketch drawn by user and sends the prediction to middle layer
- Received sketches sent by middle layer and responds in a json format
- Uses FastCGI to integrate with Apache web server
- User feedback is used to retrain the engine

Limitations

- Need at least 30 images in each object category in order to retrain the system
- Probable spammy feedback from users might hurt the system confidence
- We started with only 12 category of images because it takes more time and effort to train the system with large training data
- Predicting image can take a while (around 5 seconds on our local machines)

How “Guess a Sketch” is different

- Feedbacks from users is used to retrain the system and increases the confidence
- Retrain process is accumulative and thus retraining the system does not take much time
- Classification is open to users, we don’t limit classes of objects they can draw
- It uses SOA (Service-Oriented Architecture) which gives us the ability for us to scale up the system for larger number of users

Future Scope

- Improve the performance of guess sketch engine
- Provide Web interface and IOS app
- It can be turned into a game like solution to teach kids not only drawing but also words
- Help archaeologists identify sculptures and engravings
- Help graphic designers quickly tag their drawings

Conclusion

- Tensorflow makes image recognition and dealing with convolutional neural network much easier
- Machine learning techniques can solve many complex problems that are not easily solvable using traditional techniques
- Machine learning techniques when used correctly may solve many real-life problems that we are facing today
- Convolutional Neural Network is a really interesting technology, it gives very good predictions and can learn easier

Demo

https://youtu.be/ObUesQ_7-Qs