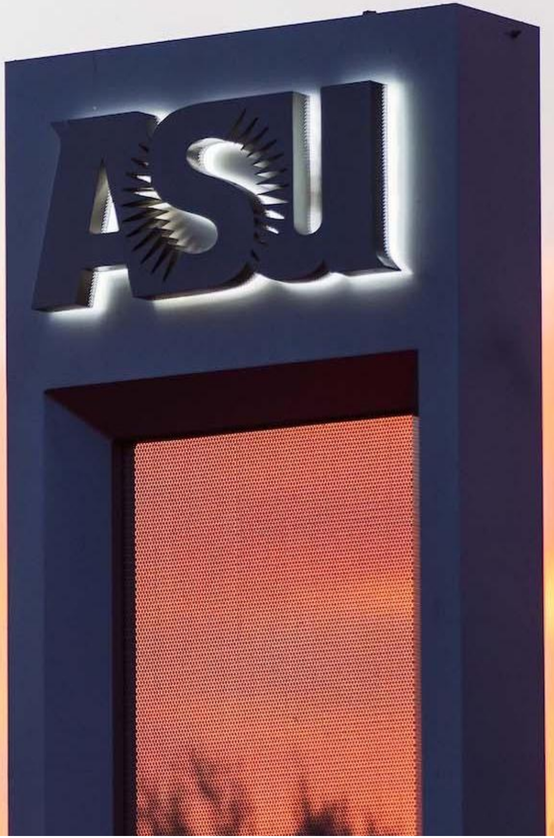
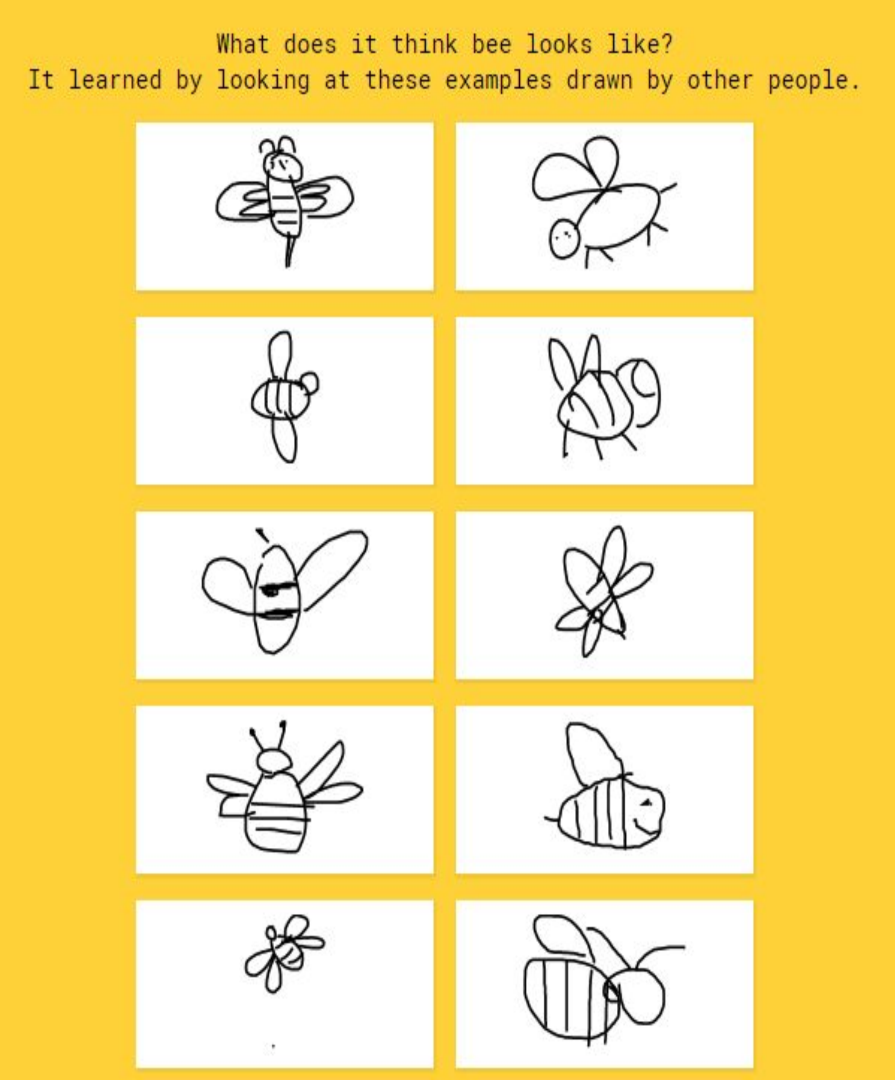
# QUICK,DRAW!
# DOODLE RECOGNITION CHALLENGE

~ THE SKETCHY GUYS!



EEE 405/591: FINAL PROJECT PRESENTATION

Arizona State University



What does it think bee looks like?
It learned by looking at these examples drawn by other people.

## PROBLEM DEFINITION:

- Released in 2018 as a kaggle challenge to educate people about AI.
- Improve pattern recognition more broadly.
- Immediate Impact on:
  - Handwriting recognition
  - OCR (Optical Character Recognition)
  - NLP (Natural Language Processing)

## CHALLENGE

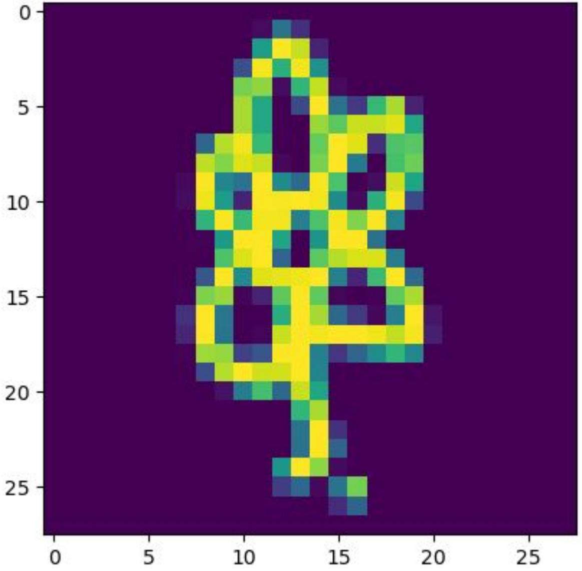
Build a recognizer to learn from noisy doodle images and perform well on a manually-labeled test set from a different distribution.

-Varshini

# OUR APPROACH

ASU Arizona State University

## DATA COLLECTION AND PRE-PROCESSING



**Flower**

Normalized Image of size 28x28

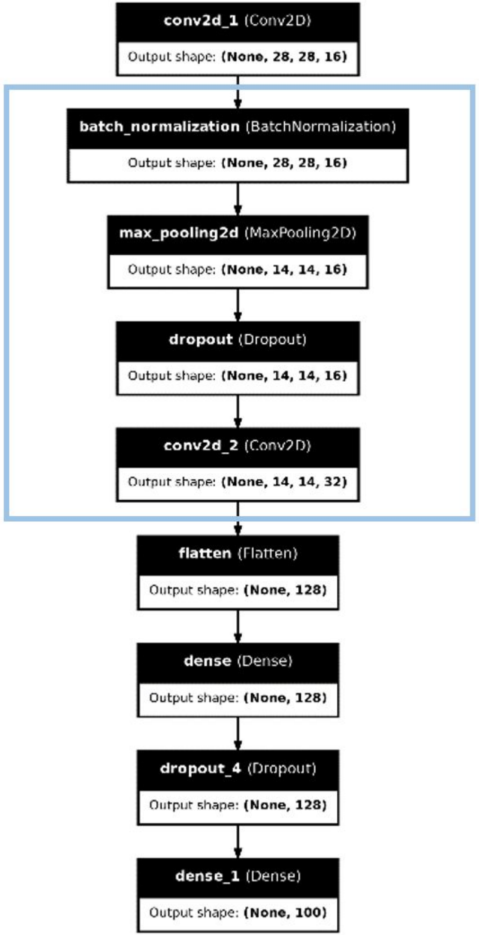
- **Data Collection** : Gather dataset of hand drawn doodles
  - 100 classes
  - Format: Numpy bitmap
  - 3968 images per class
  - Per image size : 28 X 28

- **Data Preprocessing**: 80% Train 20% Test (5 fold cross validation)
  - Train and test (X) format: (Number of samples, image-height, image-width, Number of channels)
  - Normalize (pixel_value/255.0)
  - Train and test (y) format: (One hot encoded)

Arizona State University

-Harshavardhana

# MODEL DESCRIPTION



conv2d_1 (Conv2D)
Output shape: (None, 28, 28, 16)

batch_normalization (BatchNormalization)
Output shape: (None, 28, 28, 16)

max_pooling2d (MaxPooling2D)
Output shape: (None, 14, 14, 16)

dropout (Dropout)
Output shape: (None, 14, 14, 16)

conv2d_2 (Conv2D)
Output shape: (None, 14, 14, 32)

flatten (Flatten)
Output shape: (None, 128)

dense (Dense)
Output shape: (None, 128)

dropout_4 (Dropout)
Output shape: (None, 128)

dense_1 (Dense)
Output shape: (None, 100)

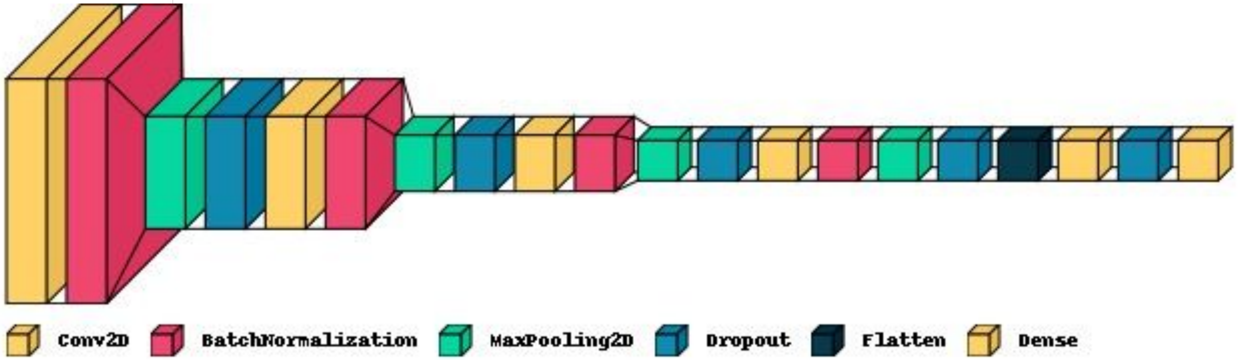
x 4

**Architecture:**

Convolutional layers : 4
Batch Normalization
ReLu activation
Max pooling and Dropout

**Model Compilation**:

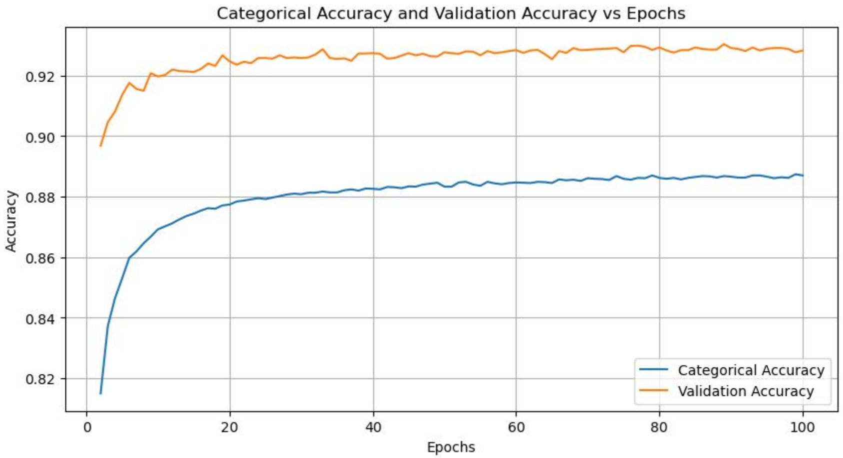
Adam optimizer
categorical cross entropy

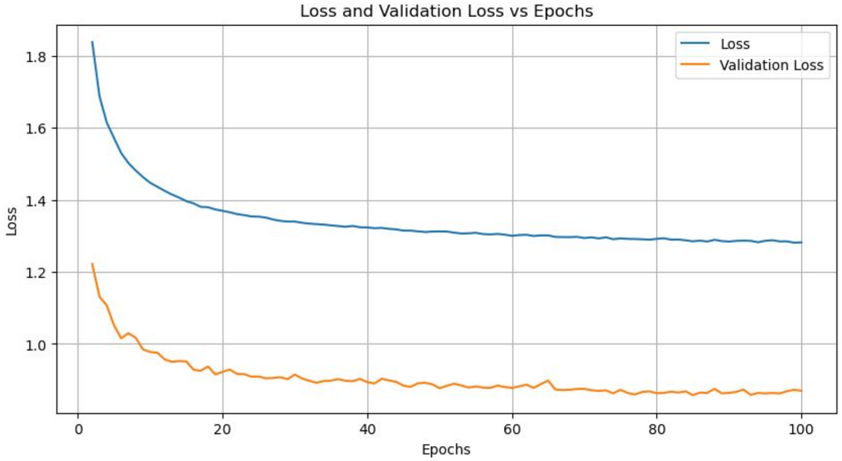


Arizona State University

Prajjwal

# MODEL TRAINING, TESTING AND EVALUATION


Categorical Accuracy and Validation Accuracy vs Epochs


Loss and Validation Loss vs Epochs

```
Epoch 100/100
4750/4750 - 103s - 22ms/step - loss: 1.2820 - top_k_categorical_accuracy:
0.8870 - val_loss: 0.8696 - val_top_k_categorical_accuracy: 0.9283
```

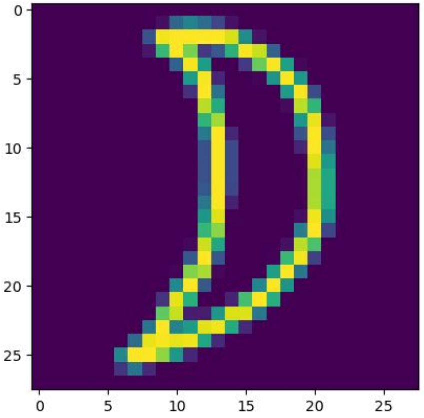
- **Model Training**:
  Validation split : 5%
  Trained with 100 epochs
  4750 steps per epoch
  64 batch size

- **Testing and Evaluation**:
  Tested on images selected
  randomly from the 20% split

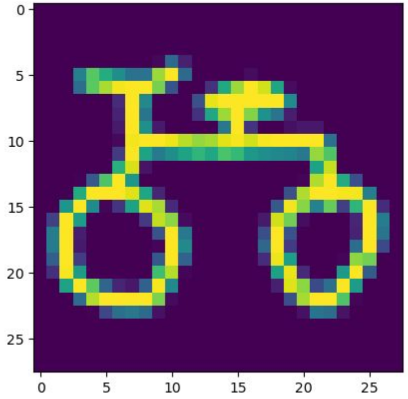
Rajdeep

# RESULTS

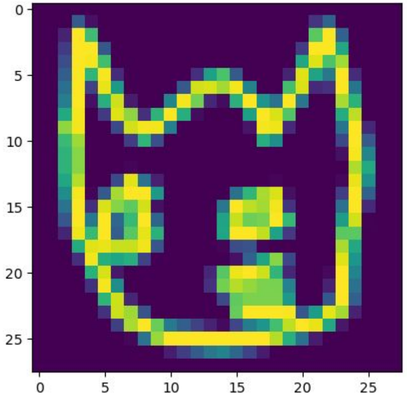
Results on few random images from the test dataset: **Top 5 categorical accuracy classes**



[**'moon'**, 'sock', 'baseball_bat', 'snake', 'knife']



[**'bicycle'**, 'headphones', 'drums', 'eyeglasses', 'dumbbell']



[**'cat'**, 'beard', 'tooth', 'alarm_clock', 'face']

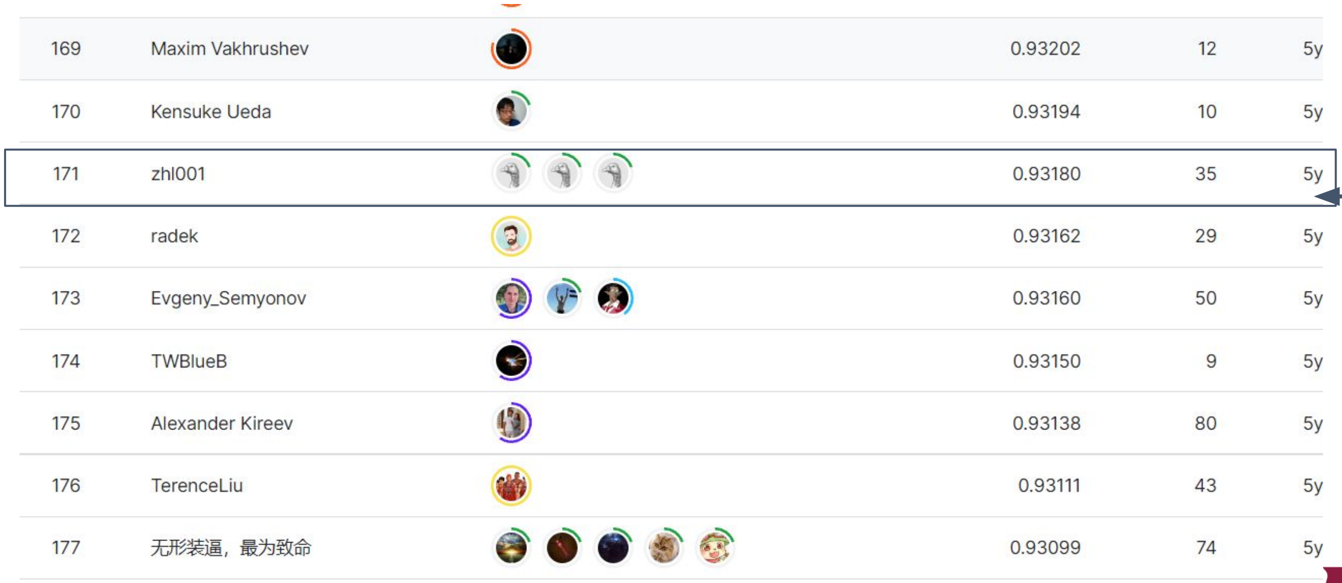
-Rajdeep

# BENCHMARK

```
score = model.evaluate(x_train, y_train, verbose=0)
print('Train accuarcy: {:0.2f}%'.format(score[1] * 100))
print('Test accuracy: {:0.2f}%'.format(score[1] * 100))
```

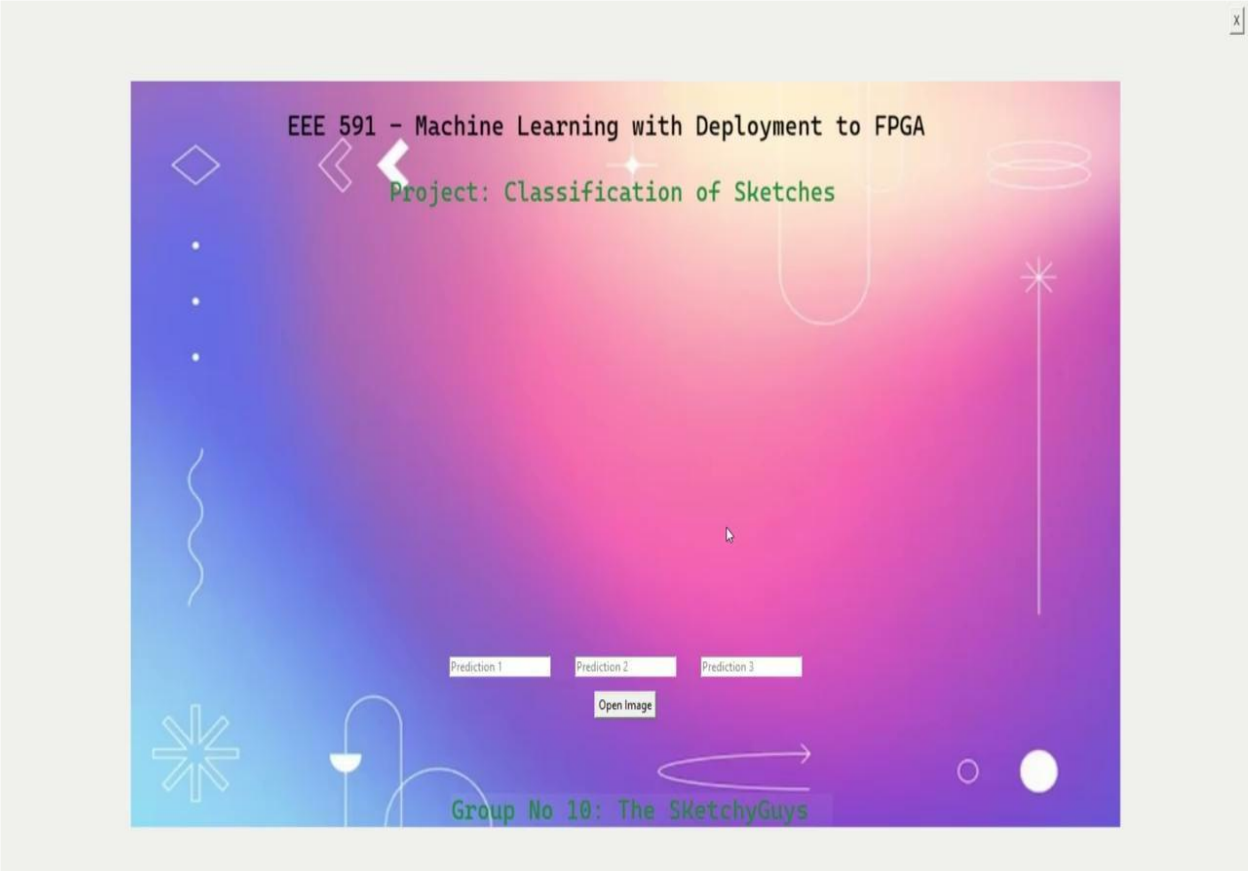
—> 93.67%

—> 93.17%



| | | | | | |
|---|---|---|---|---|---|
| 169 | Maxim Vakhrushev | | 0.93202 | 12 | 5y |
| 170 | Kensuke Ueda | | 0.93194 | 10 | 5y |
| 171 | zhl001 | | 0.93180 | 35 | 5y |
| 172 | radek | | 0.93162 | 29 | 5y |
| 173 | Evgeny_Semyonov | | 0.93160 | 50 | 5y |
| 174 | TWBlueB | | 0.93150 | 9 | 5y |
| 175 | Alexander Kireev | | 0.93138 | 80 | 5y |
| 176 | TerenceLiu | | 0.93111 | 43 | 5y |
| 177 | 无形装逼，最为致命 | | 0.93099 | 74 | 5y |

# GUI DEMONSTRATION



EEE 591 - Machine Learning with Deployment to FPGA

Project: Classification of Sketches

Group No 10: The SKetchyGuys

- Vivek

# THE SKETCHY GUYS

Harshavardhana
Varshini
Rajdeep
Prajjwal
Vivek

Arizona State University