

# Free-hand sketch classification problem

Bozitao Zhong 220030910014

Course project for SJTU CS420 Machine Learning

## 1 Main Ideas

### 1.1 Background

**QuickDraw** is one of the largest free-hand sketch datasets. It includes 345 categories of common objects, and each one contains 70 thousand training, 2.5 thousand validation and 2.5 thousand test samples. The dataset is available at [https://magenta.tensorflow.org/sketch\\_rnn](https://magenta.tensorflow.org/sketch_rnn).

In this project, we choose 25 categories (cow, panda, lion, tiger, raccoon, monkey, hedgehog, zebra, horse, owl, elephant, squirrel, sheep, dog, bear, kangaroo, whale, crocodile, rhinoceros, penguin, camel, flamingo, giraffe, pig, cat) from **QuickDraw** for the sketch classification problem. Each sketch individual is translated to a 28\*28 sketch image as the model input.

### 1.2 Transform free-hand sketch dataset into pixel image dataset

The original sketches in QuickDraw are described as vectorized sequences, which we want to further translated into sketch pixel images.

In this project, I used some functions from **pix2seq** <https://github.com/CMACH508/RPCL-pix2seq> which offers an approach to create the pixel-formed sketch images to build mine dataset transform tools `data_transform.py`. Why I rebuild that code is because that code has too much setting which are too fuzzy for me, and I'm using PyTorch rather than Tensorflow.

### 1.3 Baseline model

For baseline models, I selected simple fully-connected neural network (FCNN) as baseline. These models are relative simple for this complicated classification problem. (All baseline models are built with scikit-learn package)

### 1.4 Deep learning model

For deep learning models, I used a CNN (convolutional neural network) model for classification. As we know, CNN models are good learners in image classification tasks, like VGG and AlexNet had achieved state-of-the-art model in ImageNet task.

In this task, I adopted a model similar to VGG design, to use more 3\*3 kernels in order to improve performance.

## 2 Methods and Algorithms

### 2.1 FCNN model

A fully connected neural network is build for comparison. This model has 3 hidden layers with (100,100,50) nodes. In this project, I used this model as baseline

### 2.2 CNN model

Our CNN model is constructed based on VGG neural network. In this model, we have 3 convolutional layers, followed by 1 maxpooling layer and 3 fully connected layers. The kernel size in convolution layer is 3\*3, which is the important part in VGG.

The model structure of CNN is shown in following figure.

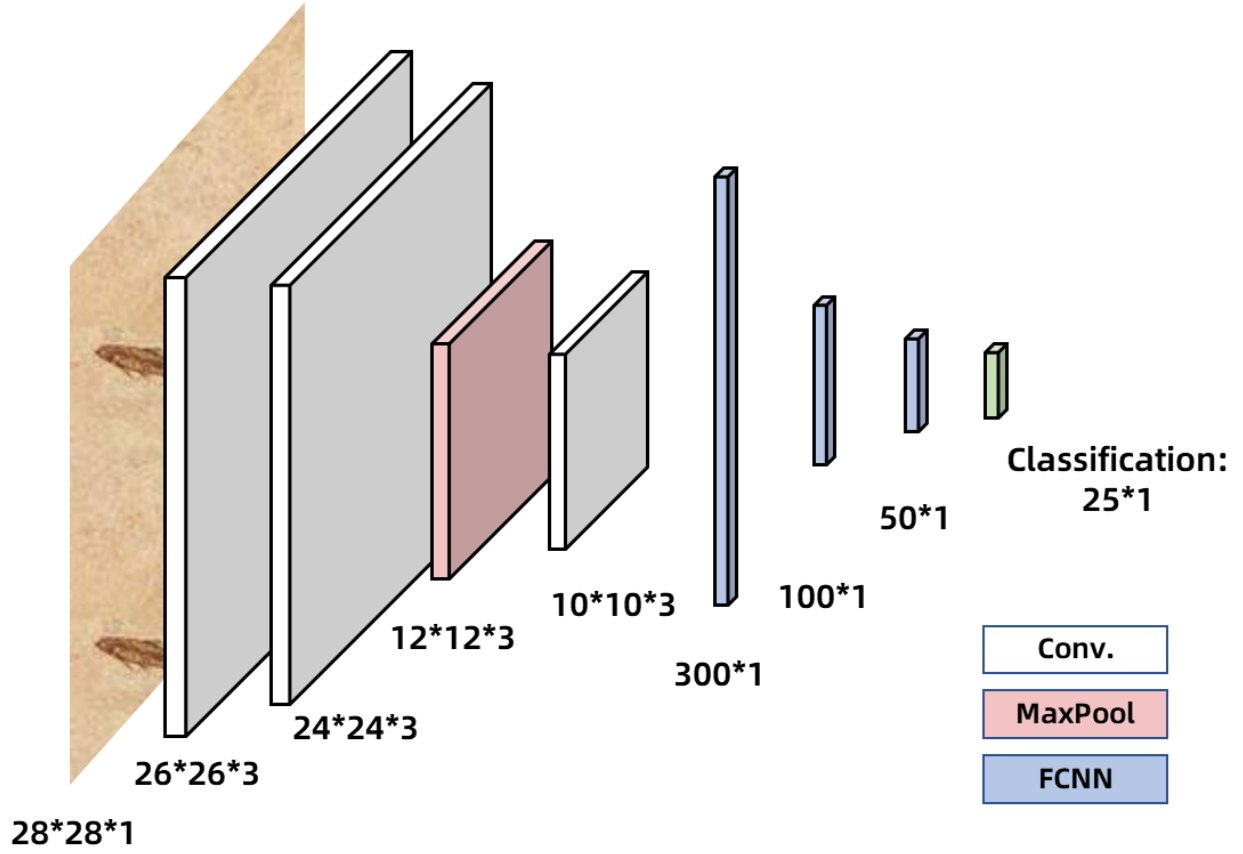


Figure 1. CNN model architecture

### 3 Experimental settings

#### 3.1 Dataset

Dataset has 25 classifications, each image is classified into 1 of the 25 classifications. Each pixel image is  $28 \times 28$  large unless have specific setting. We transformed the dataset from svg information to png images by written script [data\\_transformation.py](#). Due to the huge amount of data to be transformed, I used SJTU Siyuan HPC cluster for task parallelization.

After transformation, we have a dataset with 25 classifications, each class have a training set, a test set and a validation set. Each class's training set has 70,000  $28 \times 28$  images, test set has 2,500  $28 \times 28$  images and validation set has 2,500  $28 \times 28$  images.

#### 3.2 Training setting

In our CNN model, the learning rate is set to be 0.0005. Batch size is 100 to because it's suitable for the large dataset with  $70,000 \times 25$  images. We used accuracy, F1 score, confusion matrices as metrics for prediction results. I used 10 epoch to train the CNN model. The loss function is cross entropy.

### 4 Results

After 1 epoch of training of CNN model with the full datasets, the loss has decreased with the training process (Figure 2)

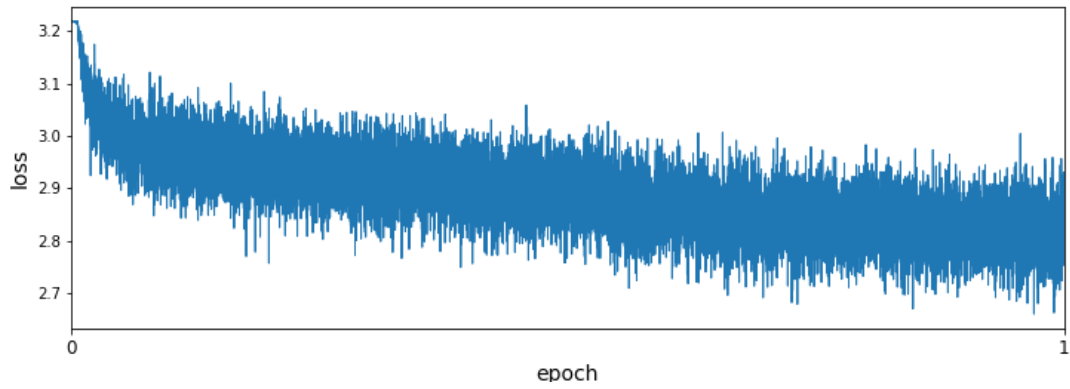


Figure 2. Loss change vs epoch

Model performance is measured by tests on both validation set and test set, the confusion matrix of these sets is shown in Figure 3 and 4.

**Validation confusion matrix epoch 1**

0	669	127	0	0	52	0	0	46	1	0	51	68	0	233	145	456	117	0	0	270	125	34	30	67	9
1	9	2152	0	0	13	0	0	18	29	0	41	5	0	11	2	8	4	0	0	62	20	33	36	24	33
2	390	200	0	0	140	0	0	10	8	0	71	65	0	340	115	407	15	0	0	123	32	99	185	284	16
3	7	220	0	0	72	0	0	15	8	0	379	11	0	62	2	55	3	0	0	328	212	18	630	16	462
4	4	11	0	0	2186	0	0	11	2	0	7	12	0	15	4	24	3	0	0	72	9	10	50	76	4
5	128	307	0	0	185	0	0	27	21	0	322	37	0	205	105	159	11	0	0	384	121	142	193	87	66
6	34	333	0	0	103	0	0	9	2	0	197	35	0	361	26	67	26	0	0	555	160	141	259	75	117
7	11	94	0	0	28	0	0	1986	122	0	14	54	0	65	8	6	22	0	0	23	9	24	6	24	4
8	1	108	0	0	4	0	0	100	2126	0	45	67	0	5	1	0	9	0	0	10	0	4	1	4	15
9	6	350	0	0	68	0	0	1	2	0	7	7	0	164	26	285	9	0	0	54	157	28	737	412	187
10	3	268	0	0	75	0	0	14	54	0	1505	13	0	13	3	16	2	0	0	184	30	15	120	33	152
11	24	68	0	0	13	0	0	75	135	0	9	1608	0	93	45	38	177	0	0	20	5	136	14	26	14
12	305	23	0	0	67	0	0	2	0	0	26	20	0	482	147	664	42	0	0	58	129	76	325	101	33
13	86	90	0	0	30	0	0	61	1	0	7	96	0	1246	81	319	37	0	0	18	29	101	192	59	47
14	47	34	0	0	10	0	0	9	0	0	4	39	0	166	1500	391	209	0	0	2	8	16	26	16	23
15	90	16	0	0	15	0	0	3	0	0	2	31	0	104	96	1944	43	0	0	5	7	24	71	34	15
16	15	27	0	0	5	0	0	22	4	0	3	121	0	81	171	35	1952	0	0	1	6	26	6	15	10
17	636	119	0	0	109	0	0	6	5	0	82	8	0	138	83	458	6	0	0	282	215	115	131	93	14
18	62	54	0	0	219	0	0	4	4	0	32	33	0	182	31	736	4	0	0	86	22	232	573	182	44
19	14	146	0	0	145	0	0	20	11	0	68	17	0	53	10	21	3	0	0	1667	96	17	42	162	8
20	26	199	0	0	34	0	0	16	3	0	42	4	0	35	1	30	0	0	0	233	1653	31	93	49	51
21	9	123	0	0	59	0	0	13	11	0	14	59	0	87	11	60	20	0	0	24	17	1760	152	43	38
22	34	73	0	0	103	0	0	1	3	0	59	11	0	86	5	196	3	0	0	61	26	46	1325	37	431
23	11	26	0	0	241	0	0	5	6	0	7	16	0	17	2	26	4	0	0	88	14	30	17	1984	6
24	0	89	0	0	20	0	0	2	10	0	117	8	0	15	3	11	1	0	0	11	8	29	472	3	1701
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figure 3. Validation set confusion matrix

Test confusion matrix epoch 1																									
0	645	109	0	0	44	0	0	49	3	0	57	55	0	226	149	447	140	0	0	267	135	47	38	74	15
1	8	2177	0	0	13	0	0	12	34	0	48	6	0	4	1	7	1	0	0	40	23	22	32	29	43
2	445	183	0	0	137	0	0	10	9	0	63	47	0	326	100	376	22	0	0	135	48	87	167	328	17
3	12	198	0	0	62	0	0	10	8	0	344	12	0	54	3	66	1	0	0	321	214	25	674	15	481
4	3	23	0	0	2166	0	0	7	1	0	9	18	0	12	1	24	4	0	0	65	10	4	52	94	7
5	111	321	0	0	191	0	0	18	22	0	338	24	0	211	90	126	19	0	0	415	127	117	231	81	58
6	40	361	0	0	93	0	0	8	1	0	175	26	0	359	17	61	24	0	0	548	172	163	272	67	113
7	19	96	0	0	19	0	0	2056	107	0	13	53	0	55	5	3	12	0	0	24	3	20	1	9	5
8	1	107	0	0	2	0	0	98	2109	0	40	69	0	6	5	1	11	0	0	14	0	3	4	7	23
9	8	361	0	0	70	0	0	1	1	0	13	9	0	161	25	286	5	0	0	58	193	22	661	433	193
10	4	259	0	0	77	0	0	10	44	0	1510	7	0	20	5	16	5	0	0	186	29	15	116	44	153
11	16	63	0	0	14	0	0	83	152	0	9	1633	0	98	24	47	194	0	0	11	6	100	15	23	12
12	296	29	0	0	62	0	0	3	0	0	21	9	0	503	127	687	48	0	0	46	140	96	322	78	33
13	97	93	0	0	26	0	0	50	1	0	0	93	0	1245	104	314	33	0	0	25	19	110	186	50	54
14	55	38	0	0	10	0	0	8	0	0	5	29	0	174	1511	382	196	0	0	3	3	16	27	20	23
15	89	12	0	0	21	0	0	6	1	0	4	17	0	146	99	1920	38	0	0	7	8	25	63	23	21
16	20	29	0	0	4	0	0	11	0	0	6	119	0	60	179	44	1965	0	0	2	6	20	5	8	22
17	632	123	0	0	130	0	0	6	3	0	79	9	0	150	92	420	2	0	0	294	218	85	123	116	18
18	62	51	0	0	219	0	0	10	6	0	33	47	0	142	35	696	3	0	0	84	24	229	593	201	65
19	25	166	0	0	135	0	0	19	6	0	94	19	0	52	5	34	0	0	0	1640	73	15	54	150	13
20	30	204	0	0	42	0	0	16	0	0	25	6	0	41	2	34	1	0	0	250	1629	16	101	45	58
21	7	128	0	0	65	0	0	11	10	0	12	64	0	76	21	58	20	0	0	20	11	1724	157	54	62
22	29	72	0	0	105	0	0	0	2	0	70	7	0	79	6	224	0	0	0	65	27	62	1283	42	427
23	13	30	0	0	231	0	0	4	4	0	7	24	0	24	3	43	3	0	0	79	13	31	13	1975	3
24	0	82	0	0	21	0	0	2	8	0	104	8	0	15	1	16	1	0	0	15	8	21	532	4	1662
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figure 4. Test set confusion matrix

The accuracy in validation set is 46.34%, in test set is 46.16%. F1 score in validation set is 0.3855, in test set is 0.3846.

From the confusion matrix, we can see that the model actually performed well in some of the classes, and some classes it didn't have a single image (like the 2nd, 3rd, 5h, 6th, 9th, 17th and 18th) classified into it.

Model	Validation Accuracy	Validation F1-score	Test Accuracy	Test F1-score
FCNN	34.61%	0.2928	34.49%	0.2906
CNN	46.34%	0.3855	46.16%	0.3846

Table 1. Performance of FCNN and CNN model

## **5 Conclusion**

In this project, I implemented 2 deep learning models to classify the free-handed sketch images. Though models are kind of simple, our classification accuracy is accepted for us. Comparing simple FCNN and CNN, convolutional layers are obviously better for image data than MLP. The test results is shown in Table 1. And some result of FCNN model is in Appendix.

The baseline SVM took me too much time for training, so I give up on training this model. This might because we have too much data point in this project.

In this course, I learned the principles and implementation of machine learning models, especially I did a lot of hands-on practices. The knowledge and skills I learned in this course will helped better in future research.

## **6 Group member and contribution**

My group only have one member: Bozitao Zhong (student ID: 220030910014), all work is done by me, including designing and building models, conduct experiments, writing report and visualization.

The code of this project is available at [https://github.com/Zuricho/Free\\_Hand\\_Sketch](https://github.com/Zuricho/Free_Hand_Sketch)

## **7 Appendix**

Validation confusion matrix epoch 1																									
0	418	193	94	41	8	1	11	92	0	100	11	0	43	97	616	214	159	0	0	61	214	51	5	62	9
1	76	1395	106	34	17	5	11	131	0	162	41	0	13	63	76	14	15	0	0	49	154	36	4	83	15
2	65	196	1077	52	37	7	25	35	0	164	36	0	49	64	212	69	52	0	0	28	35	152	25	82	38
3	12	241	96	354	20	10	27	15	0	279	77	0	60	218	39	50	4	0	0	43	266	126	116	45	402
4	10	174	48	15	1423	7	66	37	0	126	46	0	4	18	20	8	11	0	0	86	27	35	55	252	32
5	64	360	96	103	78	16	87	60	0	205	103	0	136	128	126	59	27	0	0	120	239	269	36	145	43
6	57	164	120	55	35	2	258	88	0	323	23	0	112	246	75	28	38	0	0	71	242	102	12	409	40
7	55	209	23	1	4	0	3	1729	0	6	4	0	0	56	273	10	96	0	0	0	9	8	0	11	3
8	59	395	333	0	0	0	1	1221	0	2	53	0	0	14	48	1	207	0	0	5	1	17	0	138	5
9	4	30	21	12	19	0	12	5	0	2007	1	0	73	26	82	20	13	0	0	5	15	3	21	89	42
10	11	523	209	141	49	9	74	25	0	243	431	0	35	46	40	17	12	0	0	77	92	95	39	214	118
11	76	249	228	3	1	1	14	241	0	26	56	0	17	46	235	5	1011	0	0	17	7	48	1	214	4
12	22	35	31	55	13	1	30	9	0	392	18	0	1199	154	187	80	31	0	0	12	31	56	36	46	62
13	35	117	47	37	6	0	24	104	0	195	7	0	110	694	692	109	92	0	0	7	61	49	33	30	51
14	17	25	41	1	2	0	2	24	0	90	1	0	53	77	1840	99	178	0	0	1	6	9	4	24	6
15	45	19	78	18	2	1	2	12	0	109	1	0	80	132	534	1187	136	0	0	2	9	46	13	12	62
16	14	15	12	0	3	0	0	33	0	31	0	0	22	14	484	3	1823	0	0	0	2	1	1	40	2
17	155	204	340	119	40	8	30	18	0	162	21	0	161	106	132	316	16	0	0	116	291	132	10	92	31
18	27	82	227	66	91	8	39	20	0	226	47	0	105	95	208	347	25	0	0	54	32	335	199	123	144
19	55	484	61	54	51	6	35	53	0	184	35	0	4	13	34	11	10	0	0	571	514	55	6	257	7
20	72	273	32	50	4	0	9	9	0	360	5	0	53	93	26	32	2	0	0	44	1347	21	6	38	24
21	155	154	156	56	28	2	33	79	0	47	33	0	111	152	132	182	52	0	0	88	107	800	14	99	20
22	5	66	134	148	80	9	27	5	0	277	53	0	103	114	49	111	6	0	0	24	28	165	512	32	552
23	16	100	81	5	96	1	24	14	0	193	15	0	11	7	41	6	48	0	0	247	51	28	7	1507	2
24	2	85	69	138	13	2	21	12	0	369	98	0	67	239	42	56	2	0	0	6	11	55	160	12	1041
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figure 5. Validation set confusion matrix of FCNN model

0	438	200	78	50	11	2	12	81	0	114	6	0	28	96	570	207	169	0	0	51	262	44	7	65	9
1	84	1418	111	23	15	1	11	111	0	145	60	0	15	68	64	15	7	0	0	47	164	38	2	88	13
2	71	179	1057	52	38	5	29	46	0	174	48	0	59	56	208	79	54	0	0	21	28	140	18	96	42
3	15	232	94	354	37	11	25	15	0	301	91	0	56	185	37	49	6	0	0	63	261	102	109	29	428
4	4	162	54	6	1453	7	72	39	0	110	43	0	7	22	19	9	8	0	0	97	30	41	53	230	34
5	68	380	90	109	95	23	74	61	0	223	98	0	109	142	103	67	20	0	0	138	215	240	43	144	58
6	65	190	102	61	42	0	251	82	0	297	36	0	120	230	49	24	35	0	0	76	268	126	15	390	41
7	52	183	17	1	1	0	0	1791	0	4	1	0	0	56	270	8	79	0	0	2	14	8	0	11	2
8	55	353	361	0	1	0	1	1244	0	3	47	0	1	6	40	1	194	0	0	6	2	15	0	162	8
9	7	44	22	5	14	0	4	5	0	2002	2	0	64	29	77	20	17	0	0	6	23	4	10	102	43
10	17	542	232	159	50	11	46	34	0	243	381	0	35	40	42	11	12	0	0	88	92	103	42	197	123
11	43	251	226	3	1	0	14	288	0	15	47	0	28	34	276	4	1006	0	0	9	10	57	0	180	8
12	25	30	41	53	22	2	33	7	0	413	11	0	1191	161	178	97	20	0	0	10	33	54	35	39	45
13	41	130	46	54	4	1	22	121	0	222	3	0	115	625	687	91	84	0	0	8	59	67	21	33	66
14	29	24	33	2	2	0	4	20	0	84	0	0	60	83	1855	104	166	0	0	1	0	9	3	13	8
15	45	14	75	21	0	0	2	9	0	112	5	0	107	149	5231	1192	120	0	0	0	10	53	13	13	37
16	15	11	11	0	3	0	2	35	0	35	0	0	17	17	483	0	1805	0	0	4	3	1	2	56	0
17	138	191	347	103	45	13	26	23	0	155	14	0	158	106	161	274	19	0	0	149	295	128	13	117	25
18	26	82	206	62	100	8	32	6	0	233	48	0	114	88	214	327	20	0	0	66	38	303	249	118	160
19	72	528	62	51	39	6	35	42	0	165	39	0	12	19	41	14	12	0	0	537	486	44	9	277	10
20	61	265	21	45	10	1	6	22	0	346	4	0	56	76	25	24	4	0	0	51	1390	24	8	40	21
21	166	168	148	56	33	7	29	73	0	49	31	0	90	144	152	171	44	0	0	82	98	810	36	92	21
22	13	71	156	161	82	11	23	3	0	278	53	0	107	118	37	110	6	0	0	11	21	214	464	37	524
23	22	93	83	4	105	2	36	15	0	198	11	0	8	4	37	6	35	0	0	224	55	36	11	1511	4
24	0	105	80	140	19	4	26	11	0	336	86	0	48	244	35	60	5	0	0	6	19	62	186	18	1010
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figure 6. Test set confusion matrix of FCNN model