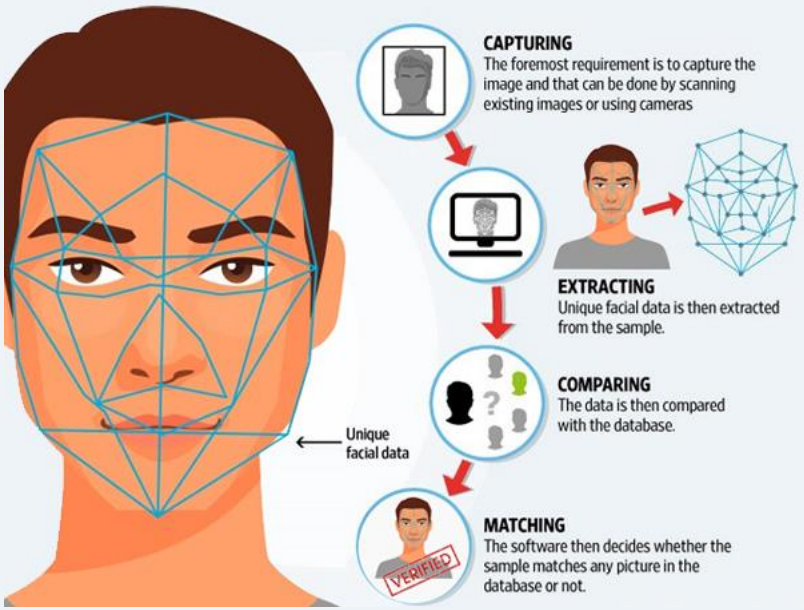
FACE RECOGNITION

Min Fang

CONTENT

Face Recognition

significant attention because of its numerous applications



Access control

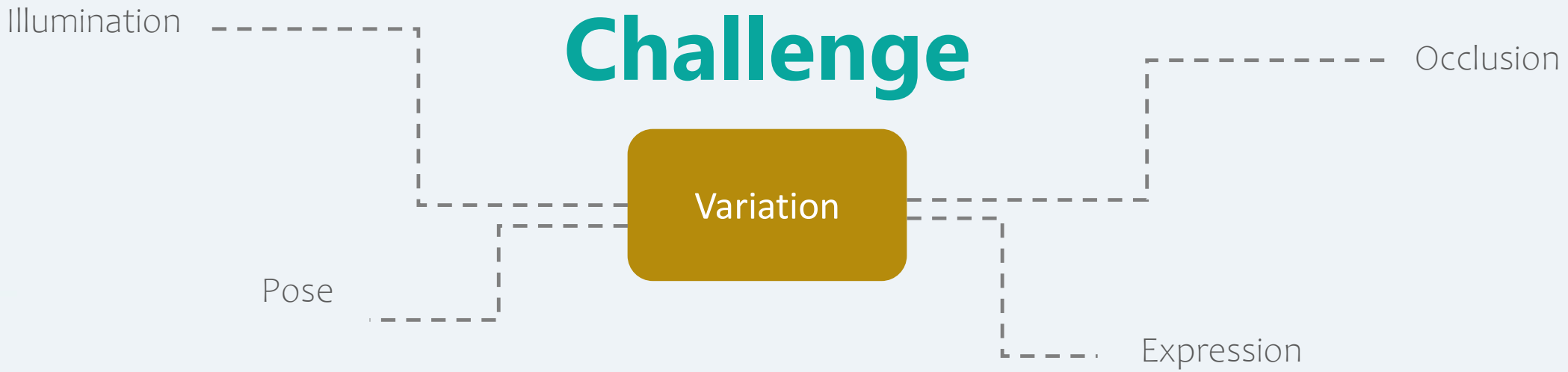
Surveillance

Security

Law enforcement

Internet communication

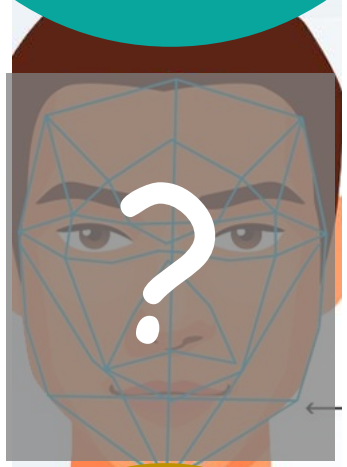
Computer entertainment


Challenge
Variation
Illumination
Pose
Occlusion
Expression

2 Objective

How precisely classify face under Illumination variation?

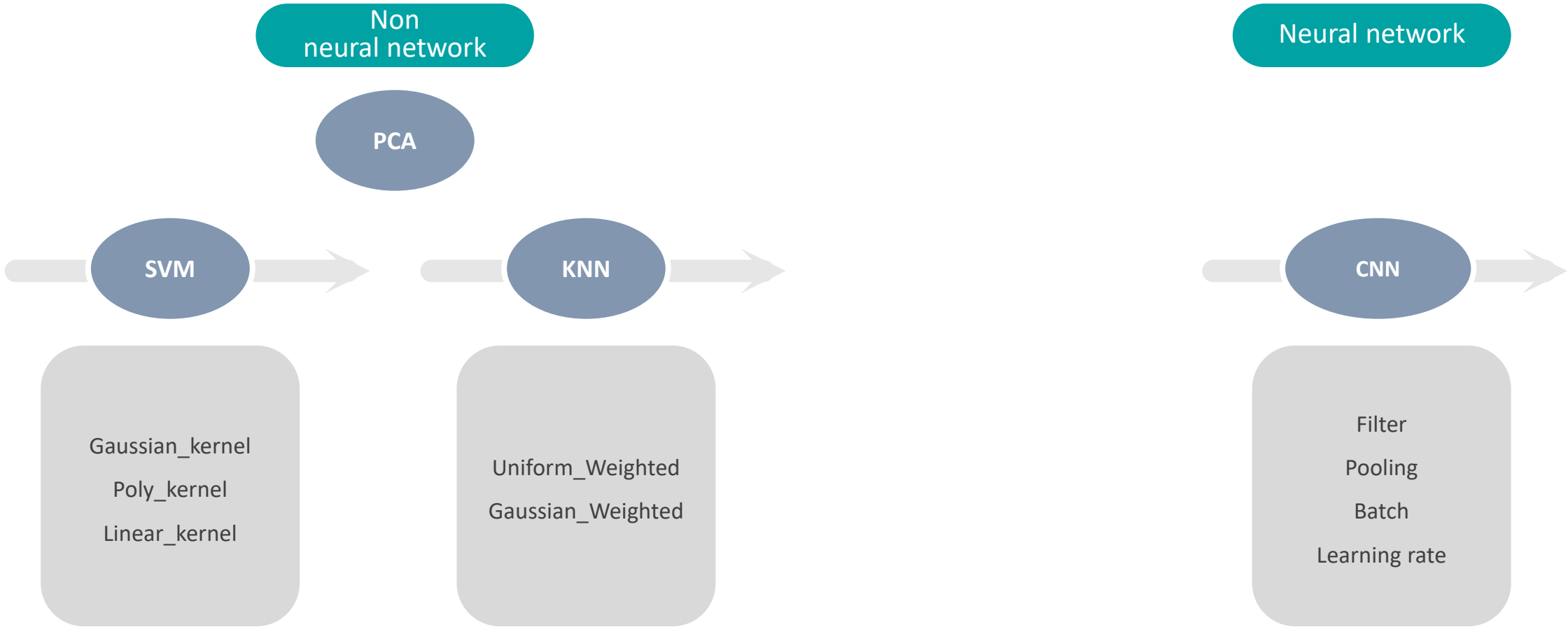
Illumination



Pose

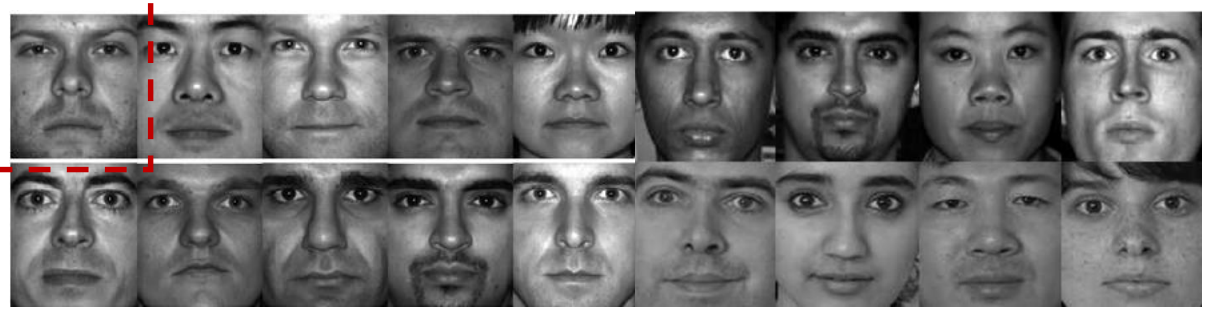
How precisely classify face under Pose variation?

3 Method Selection



4 Experiment

Data Description



Cropped version of the Yale Face Database B

UCSD. Vision
http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

192 × 168 pixels

contains 2414 images
38 different people
under
1 pose
64 illuminations

4 Experiment

Data Processing

After deleting unrelated images

Load image data from each subject folder

totally 2410 * 32256 matrix (as Row Matrix)

Split train and test data by ratio: 9 vs 1

Train: 2169

Test: 241

label	train_size			label	test_size		
01	57	21	61	01	7	21	3
02	59	22	57	02	4	22	7
03	58	23	57	03	5	23	7
04	59	24	59	04	5	24	5
05	51	25	57	05	12	25	7
06	58	26	58	06	6	26	6
07	51	27	57	07	13	27	6
08	54	28	57	08	10	28	7
09	58	29	58	09	6	29	6
10	58	30	55	10	6	30	9
11	54	31	56	11	6	31	8
12	53	32	53	12	6	32	11
13	54	33	61	13	6	33	3
15	56	34	60	15	7	34	4
16	53	35	57	16	9	35	7
17	60	36	55	17	3	36	9
18	63	37	59	19	7	37	5
19	57	38	60	20	5	38	4
20	59	39	60			39	4

4 Experiment

Non-neural network | PCA

? Number of components

140 is enough to classify for **human** eye

But

More difficult to classify for **computer**

Grid Search
(optimal PCA)

Subject 19_Eigenface

4 Experiment

Non-neural network | PCA

Number of components = 200

X_train: 2169 * 200

X_test: 241 * 200

Grid Search_SVM(gaussian)

N_FEATURES_OPTIONS = range(150,250,50)

C_OPTIONS = [1e3, 5e3, 1e4, 5e4, 1e5]

GAMMA_OPTIONS = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1]

N_FEATURES_OPTIONS = 200

4 Experiment

Non-neural network | SVM (gaussian)

Predicting people's names on the test set
done in 0.228s

	precision	recall	f1-score	support
01	1.00	1.00	1.00	7
02	1.00	1.00	1.00	4
03	1.00	1.00	1.00	5
04	1.00	1.00	1.00	5
05	1.00	1.00	1.00	12
06	1.00	1.00	1.00	6
07	0.93	1.00	0.96	13
08	1.00	1.00	1.00	10
09	1.00	1.00	1.00	6
10	1.00	1.00	1.00	6
11	0.86	1.00	0.92	6
12	1.00	1.00	1.00	6
13	1.00	1.00	1.00	6
15	1.00	1.00	1.00	7
16	0.89	0.89	0.89	9
17	1.00	1.00	1.00	3
18	0.00	0.00	0.00	0
19	1.00	1.00	1.00	7
20	0.83	1.00	0.91	5
21	1.00	1.00	1.00	3
22	1.00	1.00	1.00	7
23	1.00	0.86	0.92	7
24	1.00	1.00	1.00	5
25	1.00	0.86	0.92	7
26	1.00	1.00	1.00	6
27	1.00	1.00	1.00	6
28	1.00	1.00	1.00	7
29	1.00	0.83	0.91	6
30	1.00	1.00	1.00	9
31	1.00	1.00	1.00	8
32	1.00	0.82	0.90	11
33	1.00	1.00	1.00	3
34	1.00	1.00	1.00	4
35	0.88	1.00	0.93	7
36	0.90	1.00	0.95	9
37	1.00	1.00	1.00	5
38	1.00	0.75	0.86	4
39	1.00	1.00	1.00	4
avg / total	0.98	0.97	0.97	241

Time: 0.228s
Accuracy: 0.98

C
Gamma

decision_function_shape

Optimal_sigmma

$$\sigma = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - k\mathrm{NN}(\mathbf{x}_i)\|_2$$

- Sampling 800 points by **label proportion** with **K=7**
- compute the optimal σ **= 113.0556**, γ **= 3.9119e-05**

+ Grid Search

C = 1e5

Gamma = 3.9119e-05

Decision_function_shape = “OVR”

Ref to G. Chen’s *Simple, Fast and Accurate Hyper- parameter Tuning in Gaussian-kernel SVM*

4 Experiment

Non-neural network | SVM (poly)

```
Predicting people's names on the test set
done in 0.392s
             precision    recall  f1-score   support

         01       1.00      1.00      1.00         7
         02       1.00      1.00      1.00         4
         03       1.00      1.00      1.00         5
         04       1.00      1.00      1.00         5
         05       1.00      1.00      1.00        12
         06       1.00      1.00      1.00         6
         07       1.00      0.92      0.96        13
         08       1.00      1.00      1.00        10
         09       0.86      1.00      0.92         6
         10       0.86      1.00      0.92         6
         11       1.00      1.00      1.00         6
         12       1.00      1.00      1.00         6
         13       1.00      1.00      1.00         6
         15       1.00      1.00      1.00         7
         16       0.90      1.00      0.95         9
         17       1.00      1.00      1.00         3
         19       1.00      1.00      1.00         7
         20       1.00      1.00      1.00         5
         21       1.00      1.00      1.00         3
         22       1.00      0.86      0.92         7
         23       0.88      1.00      0.93         7
         24       1.00      1.00      1.00         5
         25       1.00      1.00      1.00         7
         26       1.00      1.00      1.00         6
         27       1.00      1.00      1.00         6
         28       0.88      1.00      0.93         7
         29       1.00      0.83      0.91         6
         30       1.00      0.89      0.94         9
         31       1.00      1.00      1.00         8
         32       1.00      0.82      0.90        11
         33       1.00      1.00      1.00         3
         34       1.00      1.00      1.00         4
         35       0.88      1.00      0.93         7
         36       1.00      1.00      1.00         9
         37       1.00      1.00      1.00         5
         38       1.00      1.00      1.00         4
         39       1.00      1.00      1.00         4

avg / total       0.98      0.98      0.97       241
```

Time: 0.392s
Accuracy: 0.98

C
Degree

decision_function_shape

Grid Search

DEGREE_OPTIONS = [2,3]

C_OPTIONS = [1e3, 5e3, 1e4, 5e4, 1e5]

C = 1e4

Degree = 2

Decision_function_shape = “OVR”

Ref to G. Chen’s *Simple, Fast and Accurate Hyper- parameter Tuning in Gaussian-kernel SVM*

4 Experiment

Non-neural network

SVM (linear)

```
Predicting people's names on the test set
done in 0.108s
             precision    recall  f1-score   support

         01       1.00      1.00      1.00         7
         02       1.00      1.00      1.00         4
         03       1.00      1.00      1.00         5
         04       1.00      1.00      1.00         5
         05       1.00      1.00      1.00        12
         06       1.00      1.00      1.00         6
         07       1.00      1.00      1.00        13
         08       1.00      1.00      1.00        10
         09       1.00      1.00      1.00         6
         10       1.00      1.00      1.00         6
         11       1.00      1.00      1.00         6
         12       1.00      1.00      1.00         6
         13       1.00      1.00      1.00         6
         15       1.00      1.00      1.00         7
         16       0.90      1.00      0.95         9
         17       0.75      1.00      0.86         3
         19       1.00      1.00      1.00         7
         20       1.00      1.00      1.00         5
         21       1.00      1.00      1.00         3
         22       1.00      1.00      1.00         7
         23       1.00      1.00      1.00         7
         24       1.00      1.00      1.00         5
         25       1.00      1.00      1.00         7
         26       1.00      1.00      1.00         6
         27       1.00      1.00      1.00         6
         28       1.00      1.00      1.00         7
         29       1.00      0.83      0.91         6
         30       1.00      0.89      0.94         9
         31       1.00      1.00      1.00         8
         32       1.00      0.91      0.95        11
         33       1.00      1.00      1.00         3
         34       1.00      1.00      1.00         4
         35       0.88      1.00      0.93         7
         36       1.00      1.00      1.00         9
         37       1.00      1.00      1.00         5
         38       1.00      1.00      1.00         4
         39       1.00      1.00      1.00         4

avg / total       0.99      0.99      0.99       241
```

Time: 0.1s
Accuracy: 0.99

C

decision_function_shape

Grid Search

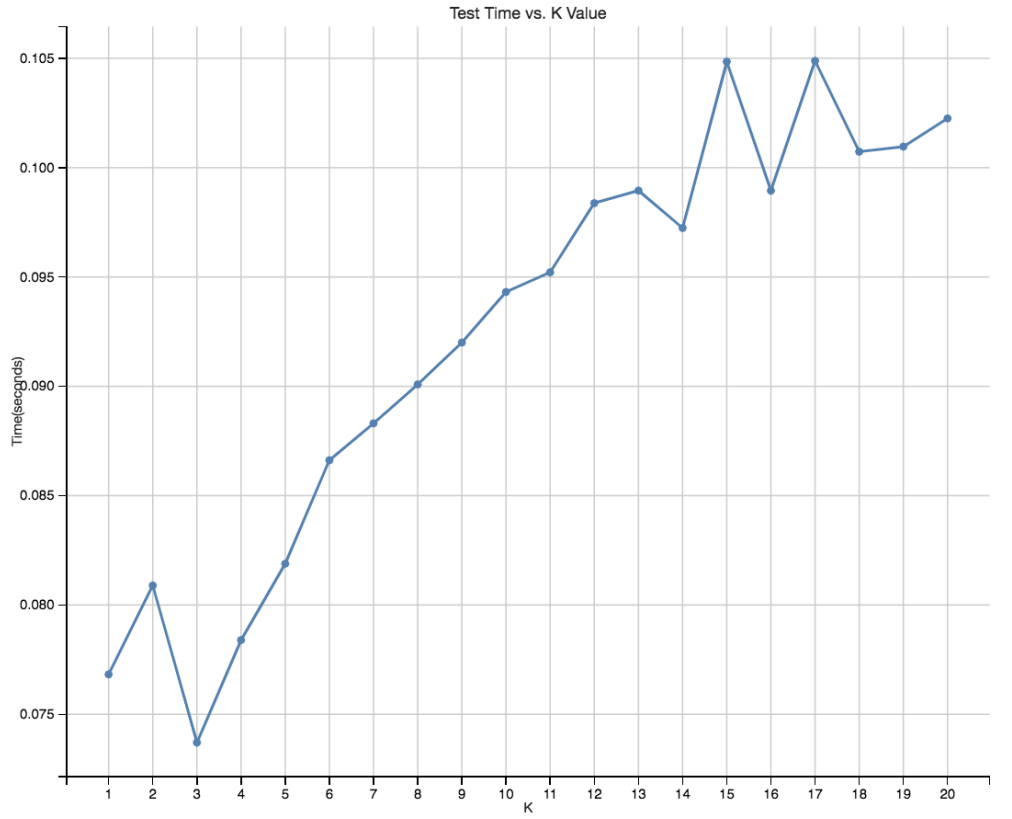
C_OPTIONS = [1e3, 5e3, 1e4, 5e4, 1e5]

C = 1e3

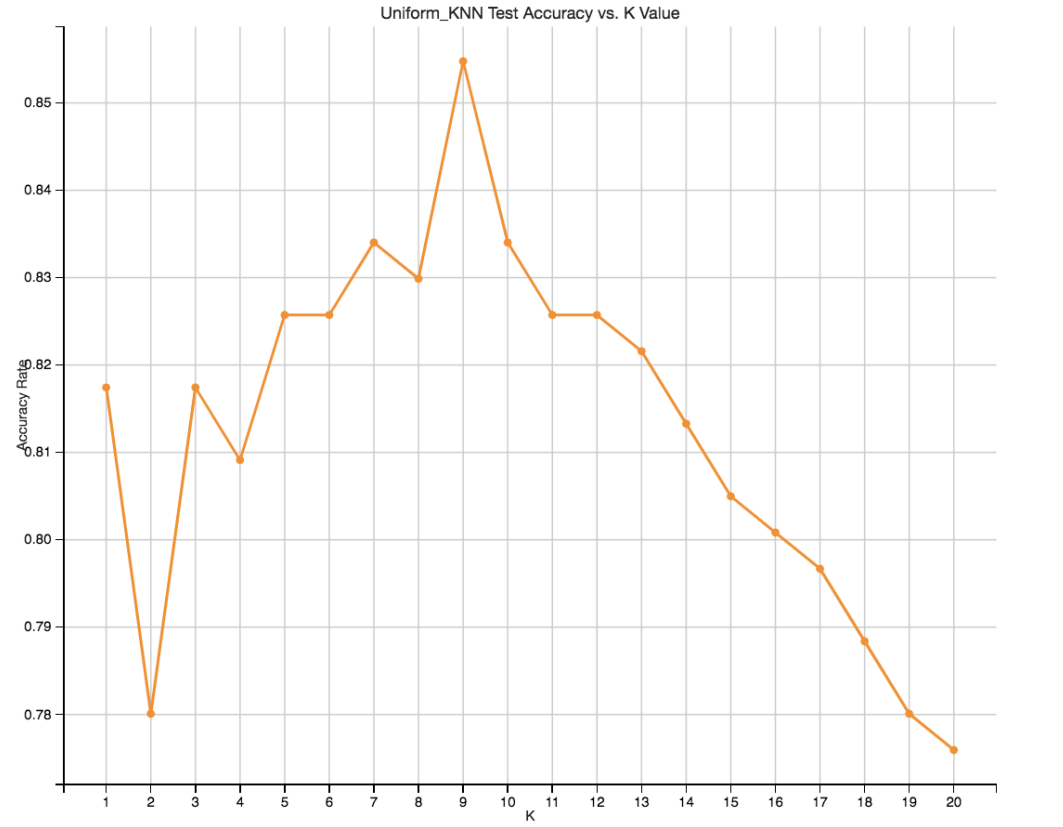
Decision_function_shape = “OVR”

4 Experiment

Non-neural network | KNN (uniform)



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0.0768	0.0809	0.0737	0.0784	0.0819	0.0866	0.0883	0.0901	0.092	0.0943	0.0952	0.0984	0.099	0.0972	0.1048	0.0989	0.1049	0.1007	0.101	0.1023



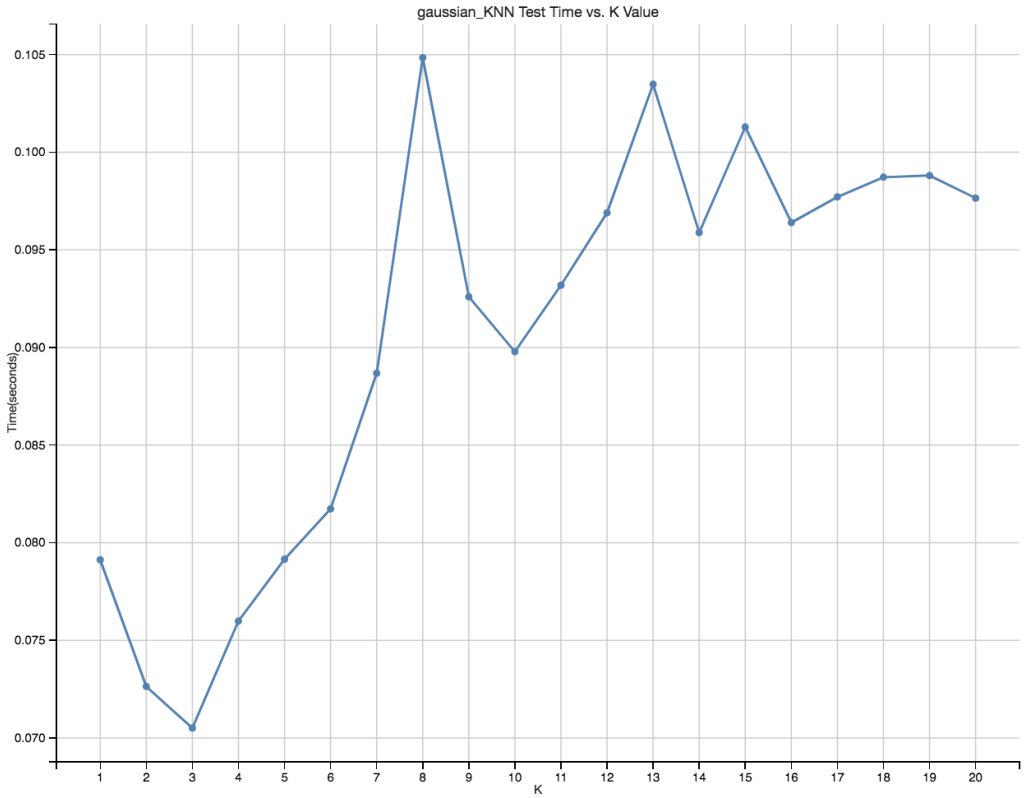
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0.8174	0.7801	0.8174	0.8091	0.8257	0.8257	0.834	0.8299	0.8548	0.8299	0.8257	0.8257	0.8299	0.8091	0.805	0.7925	0.7967	0.7884	0.7842	0.7759

Optimal K = 9
Optimal accuracy = 0.85

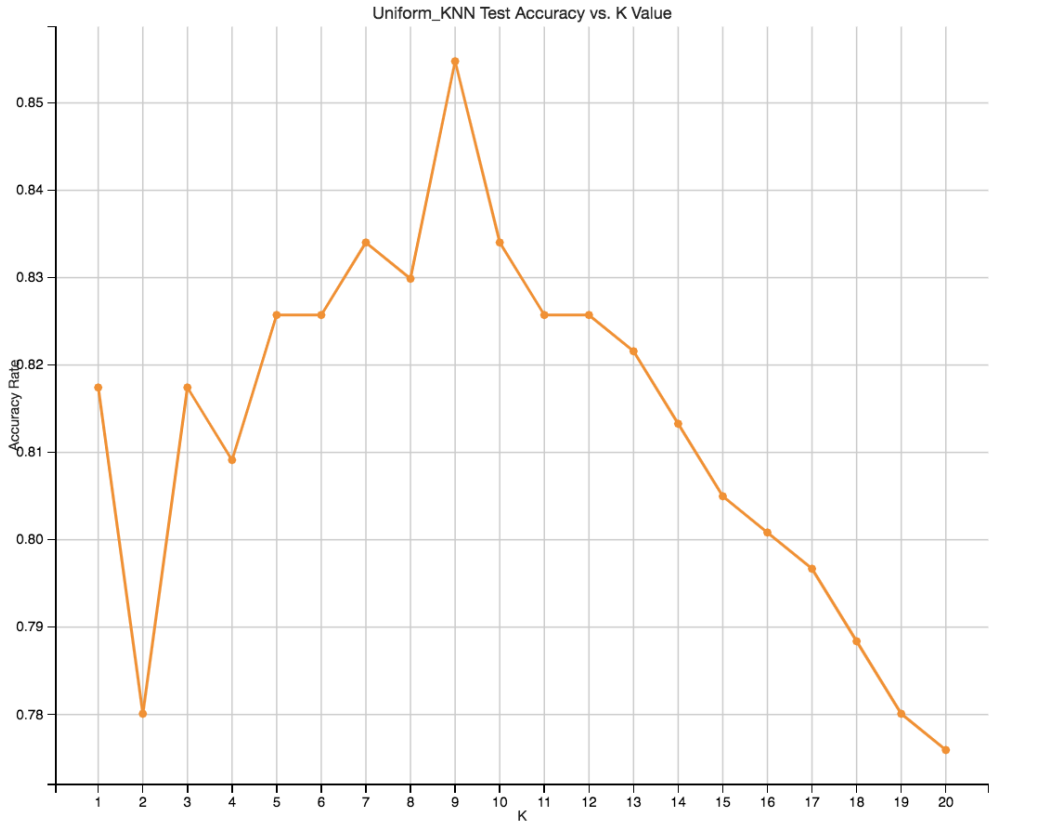
4 Experiment

Non-neural network

KNN (gaussian)



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0.0791	0.0726	0.0705	0.076	0.0792	0.0817	0.0887	0.1048	0.0926	0.0898	0.0932	0.0969	0.1035	0.0959	0.1013	0.0964	0.0977	0.0987	0.0988	0.0977

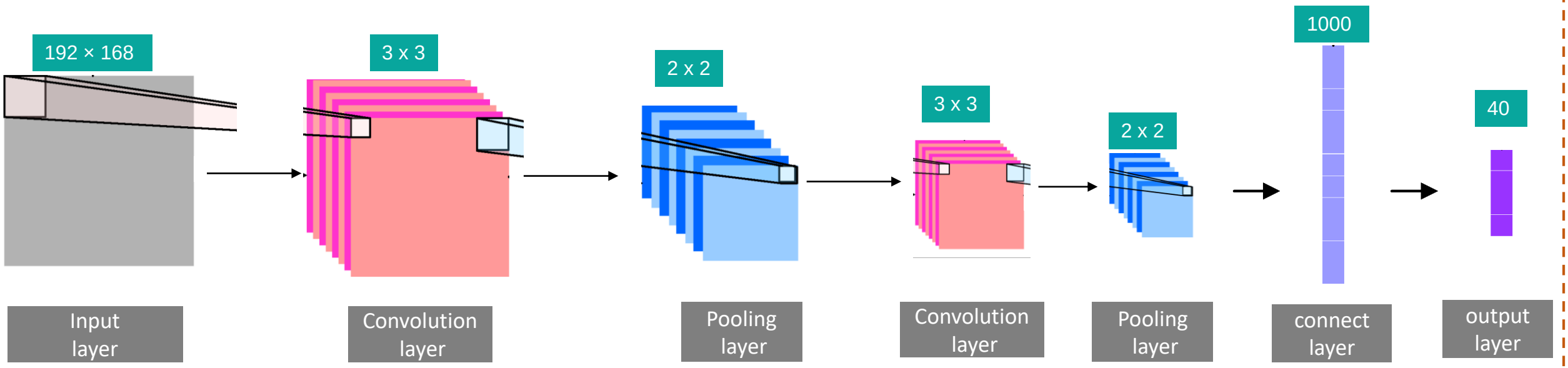


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0.8174	0.7801	0.8174	0.8091	0.8257	0.8257	0.834	0.8299	0.8548	0.8299	0.8257	0.8257	0.8299	0.8091	0.805	0.7925	0.7967	0.7884	0.7842	0.7759

Optimal K = 9
Optimal accuracy = 0.85

4 Experiment

Convolution-neural network

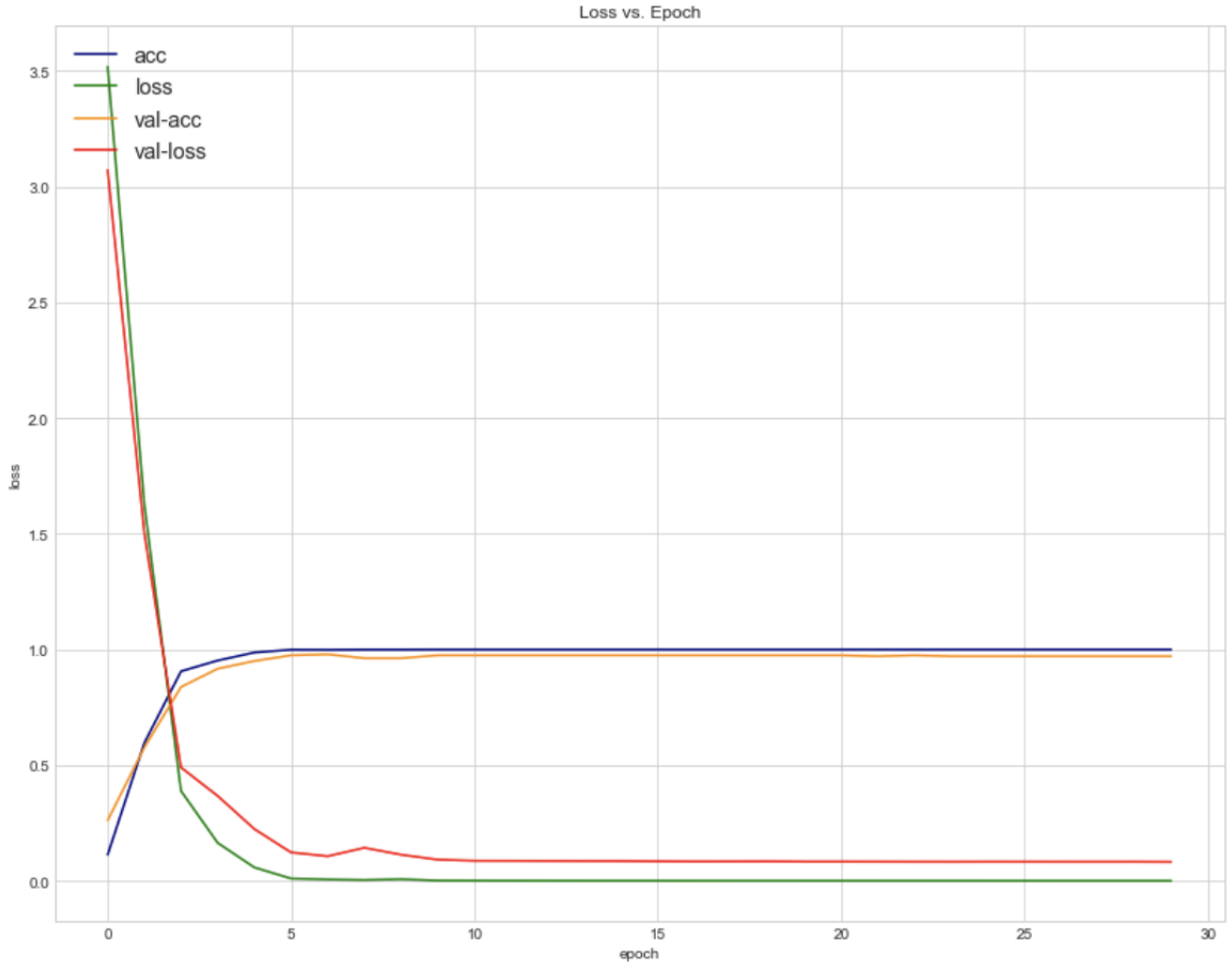


epoch = ?
batch size = 40
learning rate = ?
decay = 1e-6
Momentum = 0.9

4 Experiment

Convolution-neural network

```
Train on 2169 samples, validate on 241 samples
Epoch 1/20
 - 28s - loss: 2.8309 - acc: 0.2955 - val_loss: 2.3157 - val_acc: 0.3983
Epoch 2/20
 - 25s - loss: 0.8050 - acc: 0.7847 - val_loss: 1.0823 - val_acc: 0.6805
Epoch 3/20
 - 26s - loss: 0.2591 - acc: 0.9331 - val_loss: 0.3446 - val_acc: 0.8963
Epoch 4/20
 - 25s - loss: 0.1123 - acc: 0.9756 - val_loss: 0.9439 - val_acc: 0.7759
Epoch 5/20
 - 29s - loss: 0.1222 - acc: 0.9742 - val_loss: 0.2534 - val_acc: 0.9212
Epoch 6/20
 - 26s - loss: 0.0139 - acc: 0.9986 - val_loss: 0.1411 - val_acc: 0.9627
Epoch 7/20
 - 26s - loss: 0.0022 - acc: 1.0000 - val_loss: 0.1206 - val_acc: 0.9710
Epoch 8/20
 - 24s - loss: 0.0012 - acc: 1.0000 - val_loss: 0.1187 - val_acc: 0.9668
Epoch 9/20
 - 24s - loss: 9.5081e-04 - acc: 1.0000 - val_loss: 0.1201 - val_acc: 0.9668
Epoch 10/20
 - 24s - loss: 8.0084e-04 - acc: 1.0000 - val_loss: 0.1198 - val_acc: 0.9668
Epoch 11/20
 - 23s - loss: 6.9909e-04 - acc: 1.0000 - val_loss: 0.1196 - val_acc: 0.9668
Epoch 12/20
 - 23s - loss: 6.1535e-04 - acc: 1.0000 - val_loss: 0.1199 - val_acc: 0.9668
Epoch 13/20
 - 23s - loss: 5.5325e-04 - acc: 1.0000 - val_loss: 0.1186 - val_acc: 0.9710
Epoch 14/20
 - 23s - loss: 5.0282e-04 - acc: 1.0000 - val_loss: 0.1191 - val_acc: 0.9710
Epoch 15/20
 - 24s - loss: 4.6045e-04 - acc: 1.0000 - val_loss: 0.1196 - val_acc: 0.9710
Epoch 16/20
 - 23s - loss: 4.2575e-04 - acc: 1.0000 - val_loss: 0.1203 - val_acc: 0.9710
Epoch 17/20
 - 23s - loss: 3.9592e-04 - acc: 1.0000 - val_loss: 0.1199 - val_acc: 0.9710
Epoch 18/20
 - 23s - loss: 3.6939e-04 - acc: 1.0000 - val_loss: 0.1198 - val_acc: 0.9710
Epoch 19/20
 - 23s - loss: 3.4539e-04 - acc: 1.0000 - val_loss: 0.1200 - val_acc: 0.9710
Epoch 20/20
 - 23s - loss: 3.2489e-04 - acc: 1.0000 - val_loss: 0.1200 - val_acc: 0.9710
```

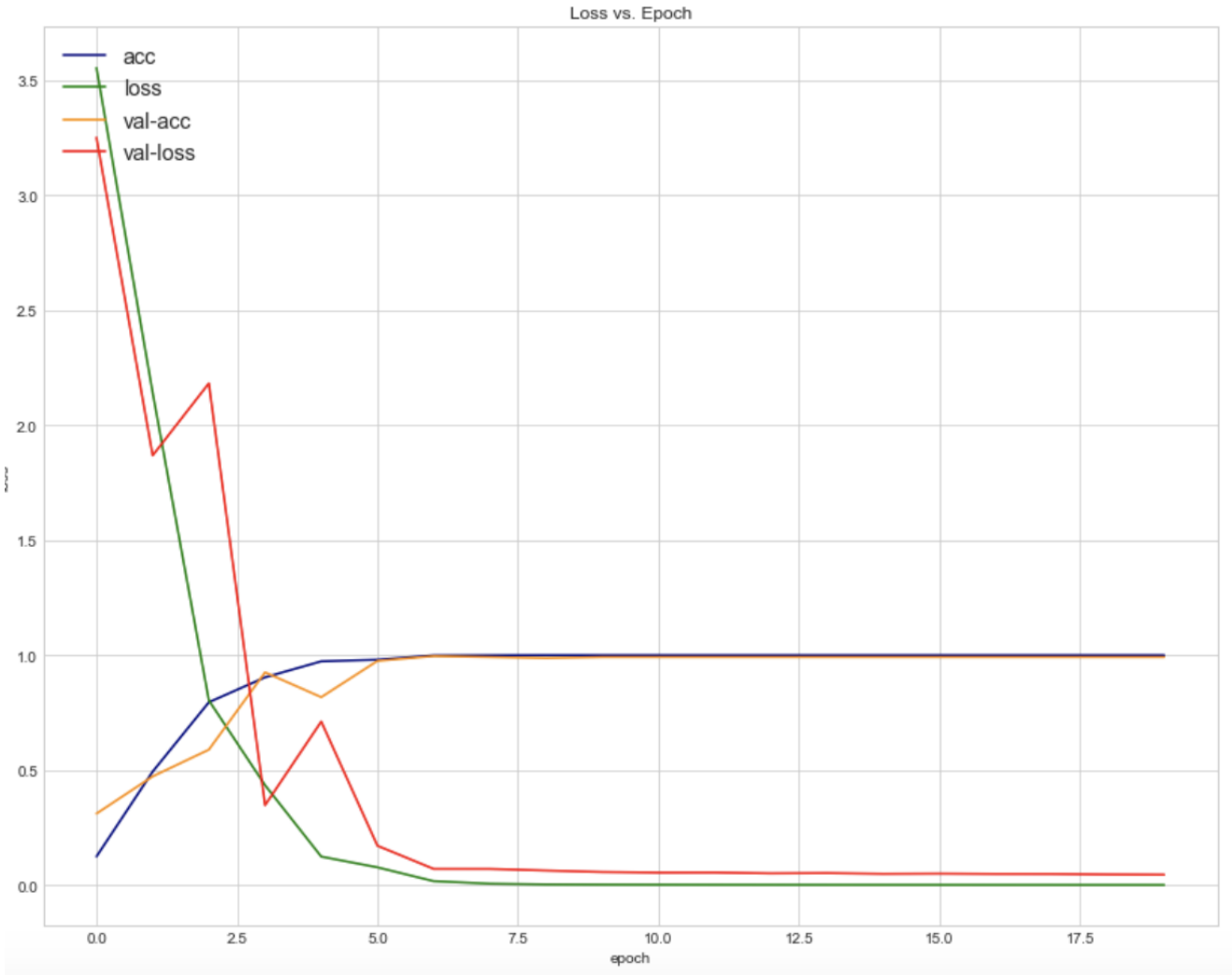


learning rate = 0.002

4 Experiment

Convolution-neural network

```
Train on 2169 samples, validate on 241 samples
Epoch 1/20
 - 25s - loss: 3.5532 - acc: 0.1245 - val_loss: 3.2501 - val_acc: 0.3112
Epoch 2/20
 - 24s - loss: 2.1410 - acc: 0.4938 - val_loss: 1.8681 - val_acc: 0.4730
Epoch 3/20
 - 23s - loss: 0.8033 - acc: 0.7953 - val_loss: 2.1818 - val_acc: 0.5892
Epoch 4/20
 - 24s - loss: 0.4346 - acc: 0.9036 - val_loss: 0.3477 - val_acc: 0.9253
Epoch 5/20
 - 26s - loss: 0.1243 - acc: 0.9733 - val_loss: 0.7114 - val_acc: 0.8174
Epoch 6/20
 - 27s - loss: 0.0776 - acc: 0.9802 - val_loss: 0.1716 - val_acc: 0.9751
Epoch 7/20
 - 25s - loss: 0.0176 - acc: 0.9991 - val_loss: 0.0709 - val_acc: 0.9959
Epoch 8/20
 - 25s - loss: 0.0065 - acc: 0.9995 - val_loss: 0.0707 - val_acc: 0.9917
Epoch 9/20
 - 28s - loss: 0.0033 - acc: 1.0000 - val_loss: 0.0637 - val_acc: 0.9876
Epoch 10/20
 - 25s - loss: 0.0025 - acc: 1.0000 - val_loss: 0.0574 - val_acc: 0.9917
Epoch 11/20
 - 26s - loss: 0.0022 - acc: 1.0000 - val_loss: 0.0547 - val_acc: 0.9917
Epoch 12/20
 - 25s - loss: 0.0019 - acc: 1.0000 - val_loss: 0.0550 - val_acc: 0.9917
Epoch 13/20
 - 26s - loss: 0.0017 - acc: 1.0000 - val_loss: 0.0515 - val_acc: 0.9917
Epoch 14/20
 - 25s - loss: 0.0015 - acc: 1.0000 - val_loss: 0.0527 - val_acc: 0.9917
Epoch 15/20
 - 26s - loss: 0.0014 - acc: 1.0000 - val_loss: 0.0495 - val_acc: 0.9917
Epoch 16/20
 - 26s - loss: 0.0013 - acc: 1.0000 - val_loss: 0.0505 - val_acc: 0.9917
Epoch 17/20
 - 26s - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0491 - val_acc: 0.9917
Epoch 18/20
 - 25s - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0488 - val_acc: 0.9917
Epoch 19/20
 - 26s - loss: 0.0010 - acc: 1.0000 - val_loss: 0.0470 - val_acc: 0.9917
Epoch 20/20
 - 25s - loss: 9.7845e-04 - acc: 1.0000 - val_loss: 0.0459 - val_acc: 0.9917
506.5965440273285 s
```



learning rate = 0.001

5 Conclusion

Discussion

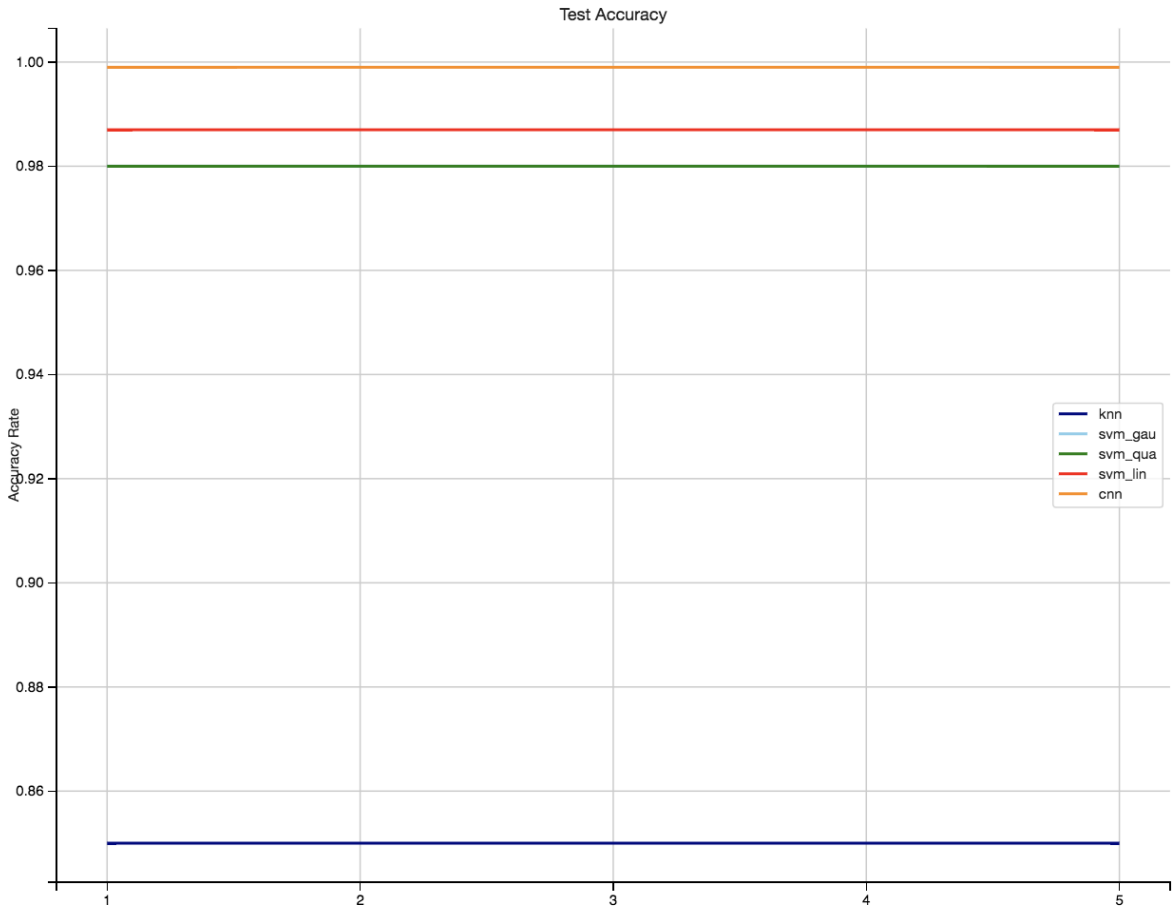
Method	Time (s)	Accuracy
KNN(K=9)	0.09	0.85
SVM (Gaussian)	0.23	0.98
SVM (poly=2)	0.39	0.98
SVM (linear)	0.10	0.987
CNN (lr=0.001)	502	0.999

Same pose under Illumination variation

Convolution Neural Network performs **best** with the highest test accuracy **0.999**

KNN performs **worst** with the lowest test accuracy **0.85**

There's **no significant difference** among the choices of kernel function choices for **SVM**



How about
Pose & Illumination Variation
?

5 Conclusion

Extra Finding

Extended Yale Face Database B

UCSD. Vision
http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

contains 16128 images of 28 human subjects
under 9 poses and 64 illumination conditions

640 × 480 pixels

Based on these experiments and data:

Cropped data VS Extended data

- more pictures as training >>> easier to classify a subject, even adding pose variation

7 subject VS 12 subjects (extended data)

- less subjects >>> easier to classify

7 subjects :
11/12/13/15/16/17/18

Method	Time	Accuracy
KNN(K=2)	0.09	1
SVM (Gaussian)	0.19	1
SVM (poly=2)	0.754	1
SVM (linear)	0.94	1
CNN	821	1

12 subjects :
11/12/13/15/16/17/18/19/23/25/29/36

Method	Time	Accuracy
KNN(K=1)	0.17	0.93
SVM (Gaussian)	0.19	0.98
SVM (poly=3)	4.77	0.99
SVM (linear)	0.75	0.98
CNN		0.99

6 Reference

Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J., "*From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose,"* IEEE Trans. Pattern Anal. Mach. Intelligence, 2001, Vol.23, pp.643-660

M.Hassaballah, Saleh Aly, "*Face recognition: challenges, achievements and future directions,"* IET Computer Vision, 2015, Vol. 9, Iss. 4, pp. 614–626

B. K. Gunturk, A. U. Batur, *"Eigenface-domain super-resolution for face recognition,"* IEEE Signal Processing Society, 2003, Vol. 12, Iss.5, pp. 597-606

V. Tata, *"Simple Image Classification using Convolutional Neural Network," https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8*

THANKS