Compare Various Classification Algorithms Using MNIST

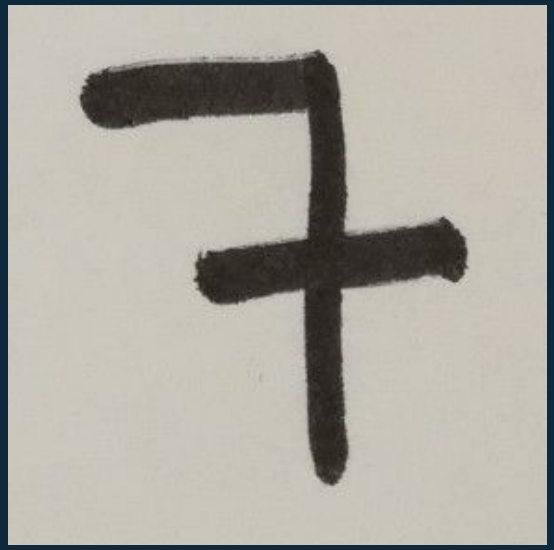
Group 8

Outline

1. Topic
2. Dataset
3. Purpose
4. Method
5. Detail of the method
6. Result & Conclusion

1 Topic

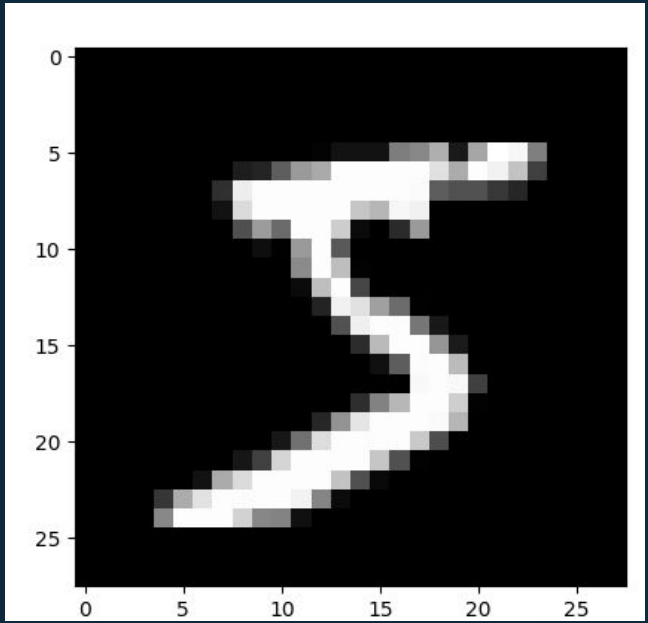
Topic

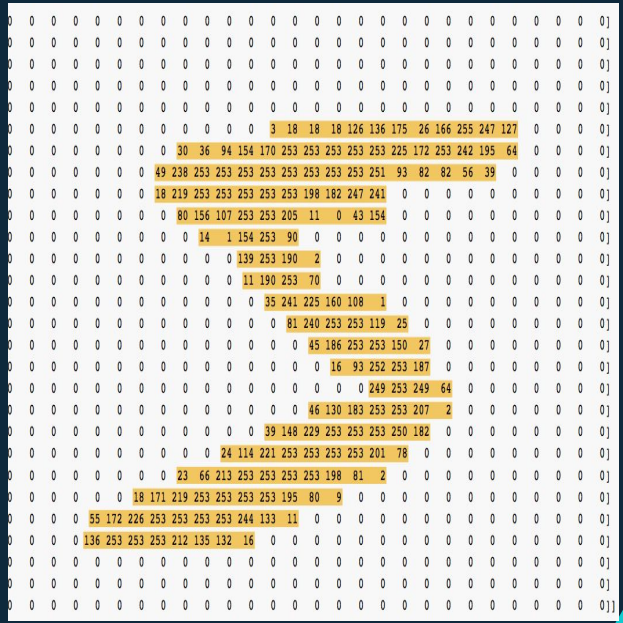


7

2 Dataset

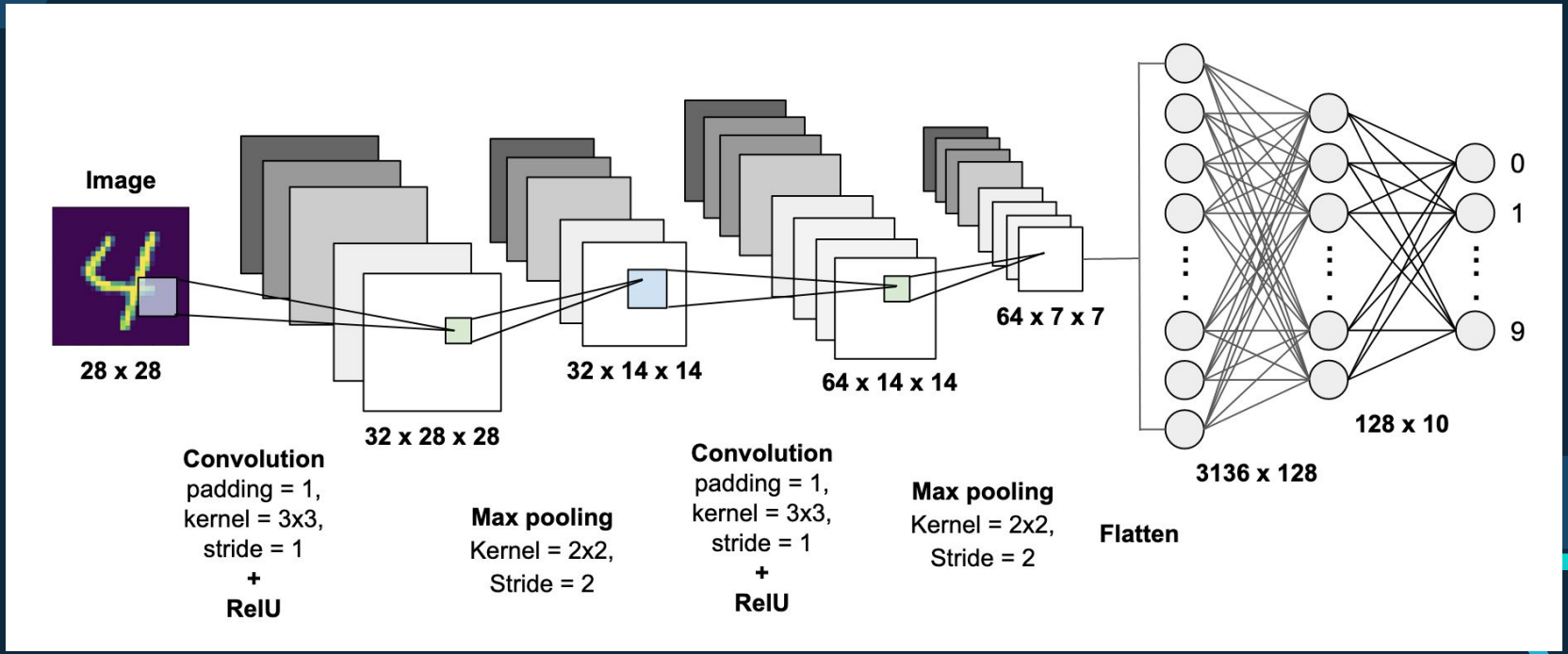
Dataset



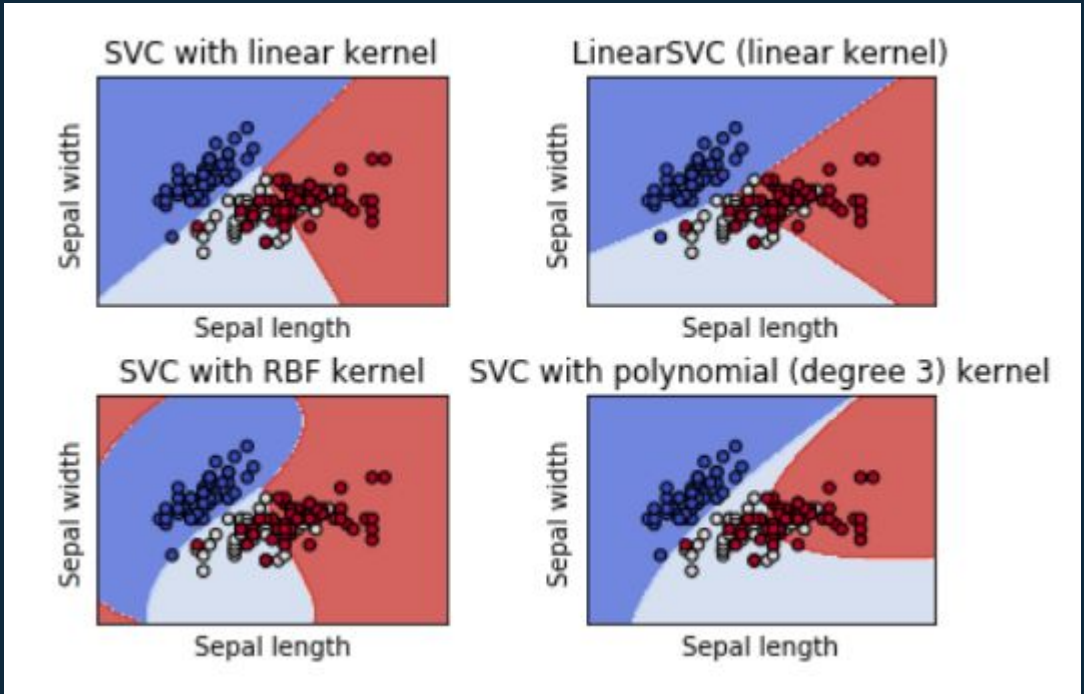


3 Purpose

Mainstream algorithm–CNN



Analyzing the algorithm



Compare using resources

Rank	Model	Percentage error	Accuracy	Trainable Parameters	Paper	Code	Result	Year
1	**Homogeneous ensemble with Simple CNN**	0.09	99.91		An Ensemble of Simple Convolutional Neural Network Models for MNIST Digit Recognition			2020
2	**Branching/Merging CNN + Homogeneous Vector Capsules**	0.13	99.87	1,514,187	No Routing Needed Between Capsules			2020
3	**EnsNet** (Ensemble learning in CNN augmented with fully connected subnetworks)	0.16	99.84		Ensemble learning in CNN augmented with fully connected subnetworks			2020
4	**Efficient-CapsNet**	0.16	99.84	161,824	Efficient-CapsNet: Capsule Network with Self-Attention Routing			2021
5	**SOPCNN** (Only a single Model)	0.17	99.83	1,400,000	Stochastic Optimization of Plain Convolutional Neural Networks with Simple methods			2020
6	**RMDL** (30 RDLs)	0.18	99.82		RMDL: Random Multimodel Deep Learning for Classification			2018

4 Method

Method

1. KNN
2. Random forest
3. SVM
    a. linear
    b. poly
    c. rbf
    d. sigmoid

Compare parameters

1. F1 score
2. Training time
3. Predict time

5 Detail of the method

Environment

- OS: Linux Ubuntu 18.04.6 LTS
- Python 3.9.6
- scikit-learn 1.0.2
- Hardware
  - Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz
  - NVIDIA GeForce GTX 1080 Ti

Random Forest

Random Forest = n * Decision Tree

Random Forest

n_estimators

criterion

Random Forest

n_estimators	criterion
10	gini
1000	entropy

Random Forest

criterion

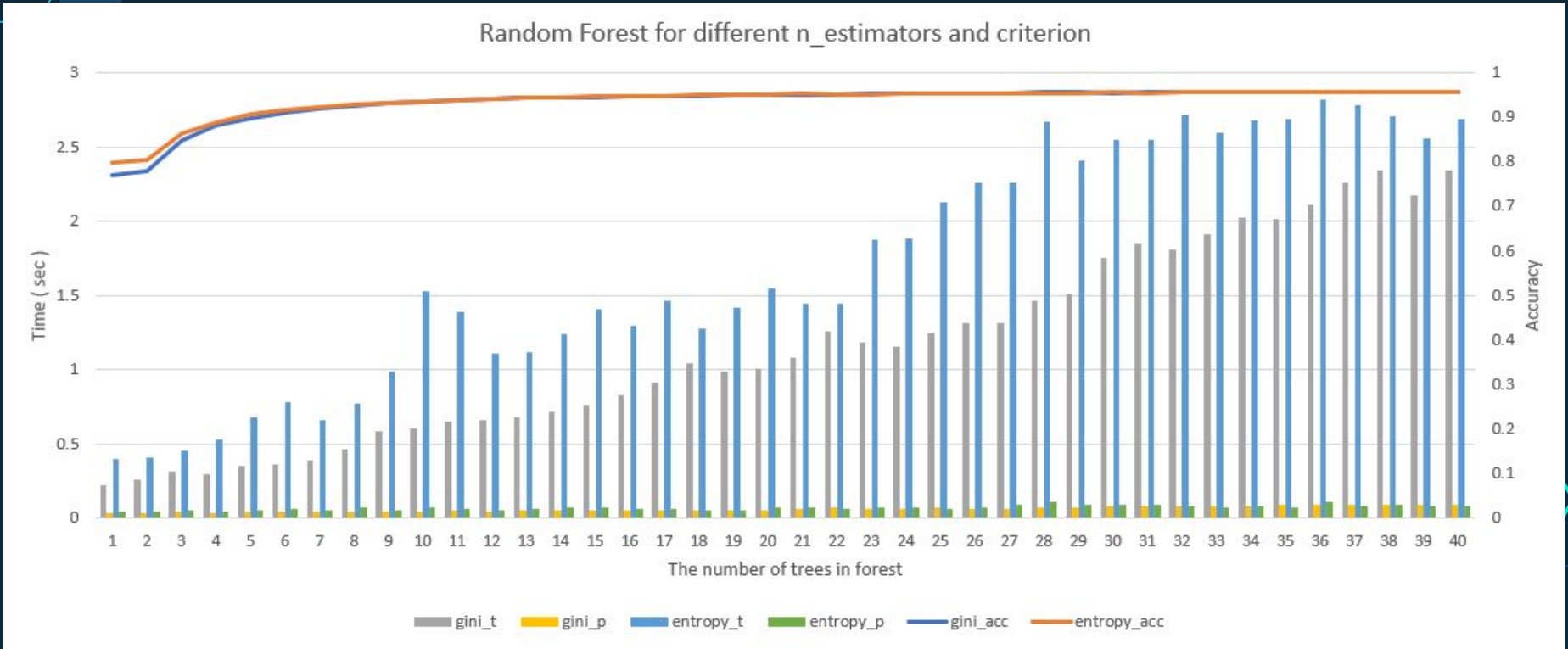
gini

$$GiniIndex = 1 - \sum_j p_j^2$$

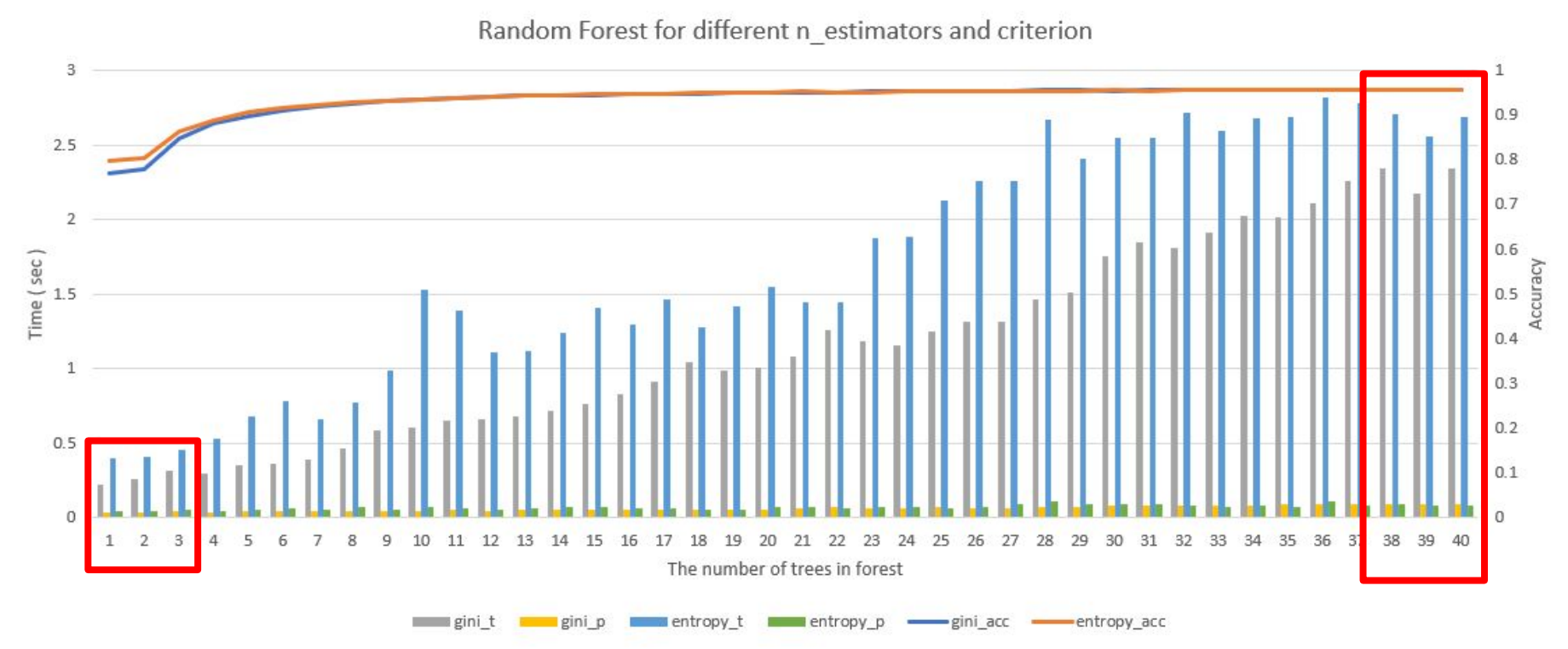
entropy

$$Entropy = -\sum_j p_j \cdot log_2 \cdot p_j$$

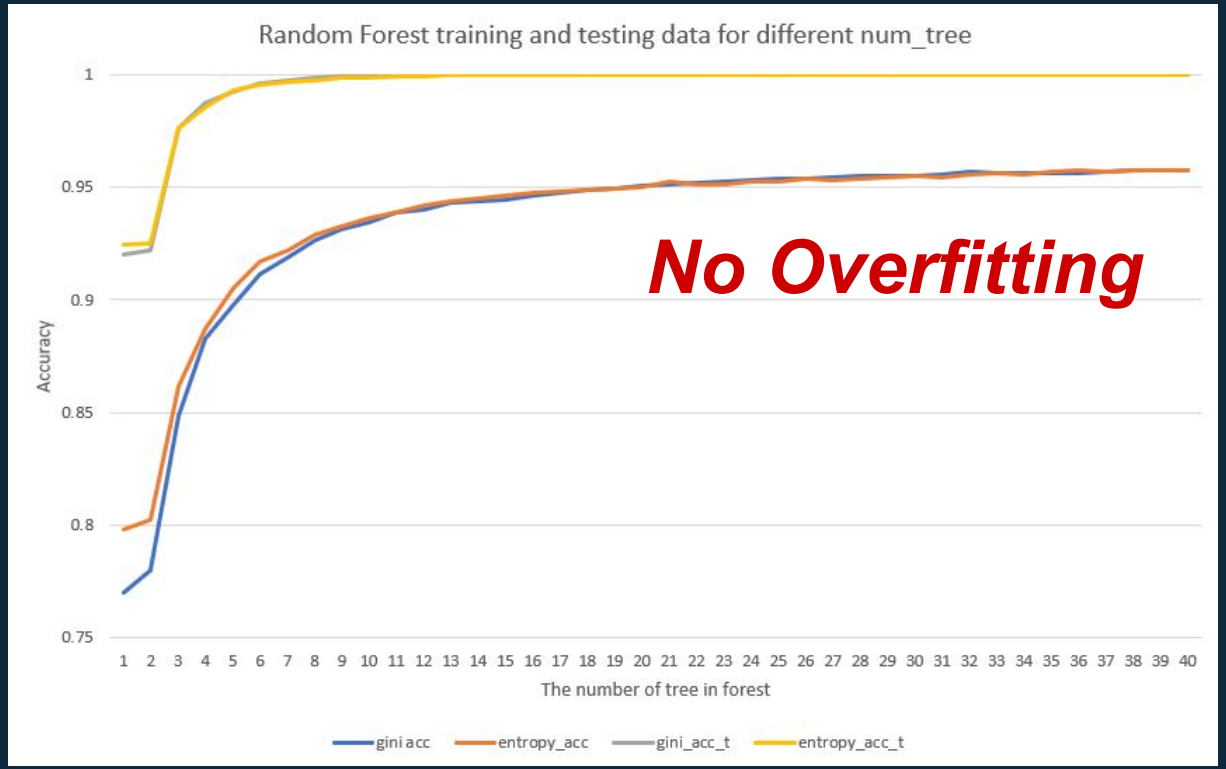
Random Forest



Random Forest



Random Forest



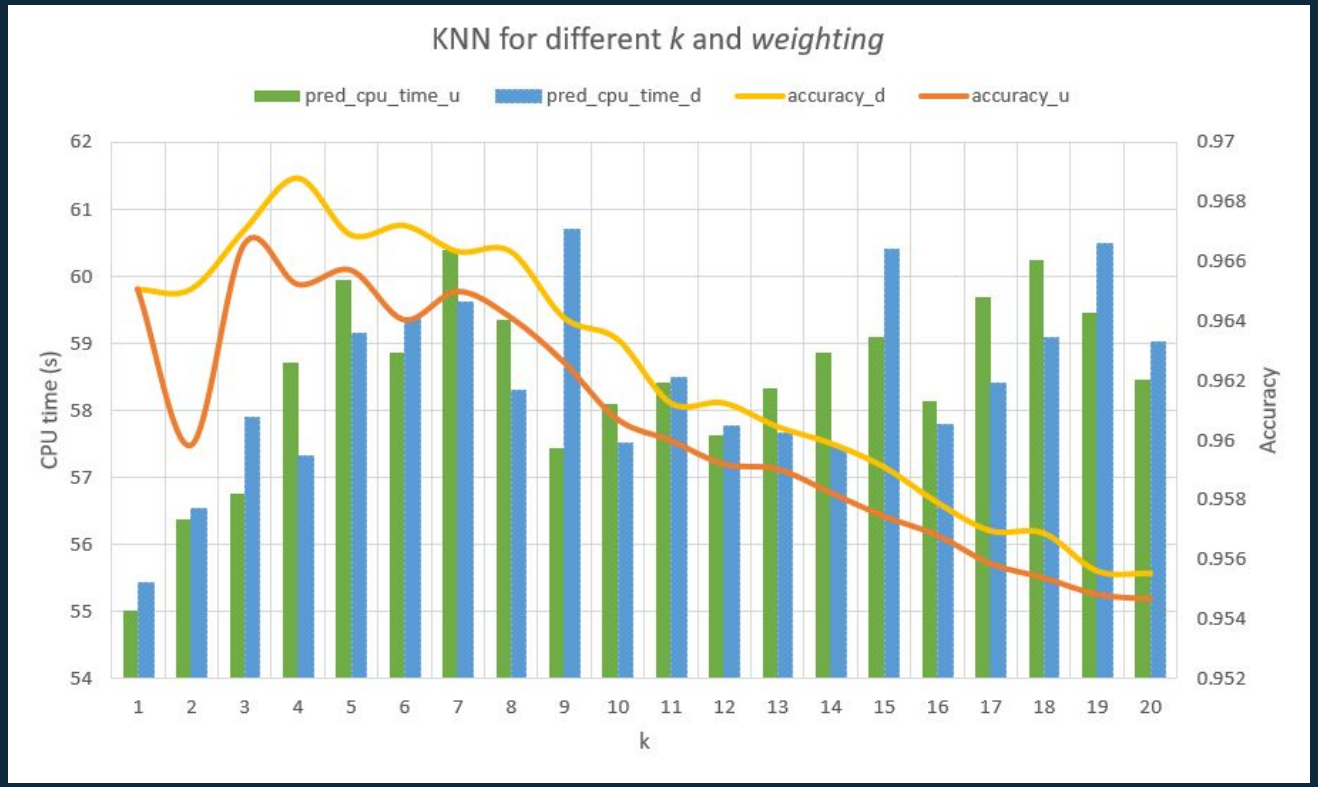
KNN

K Nearest Neighbor

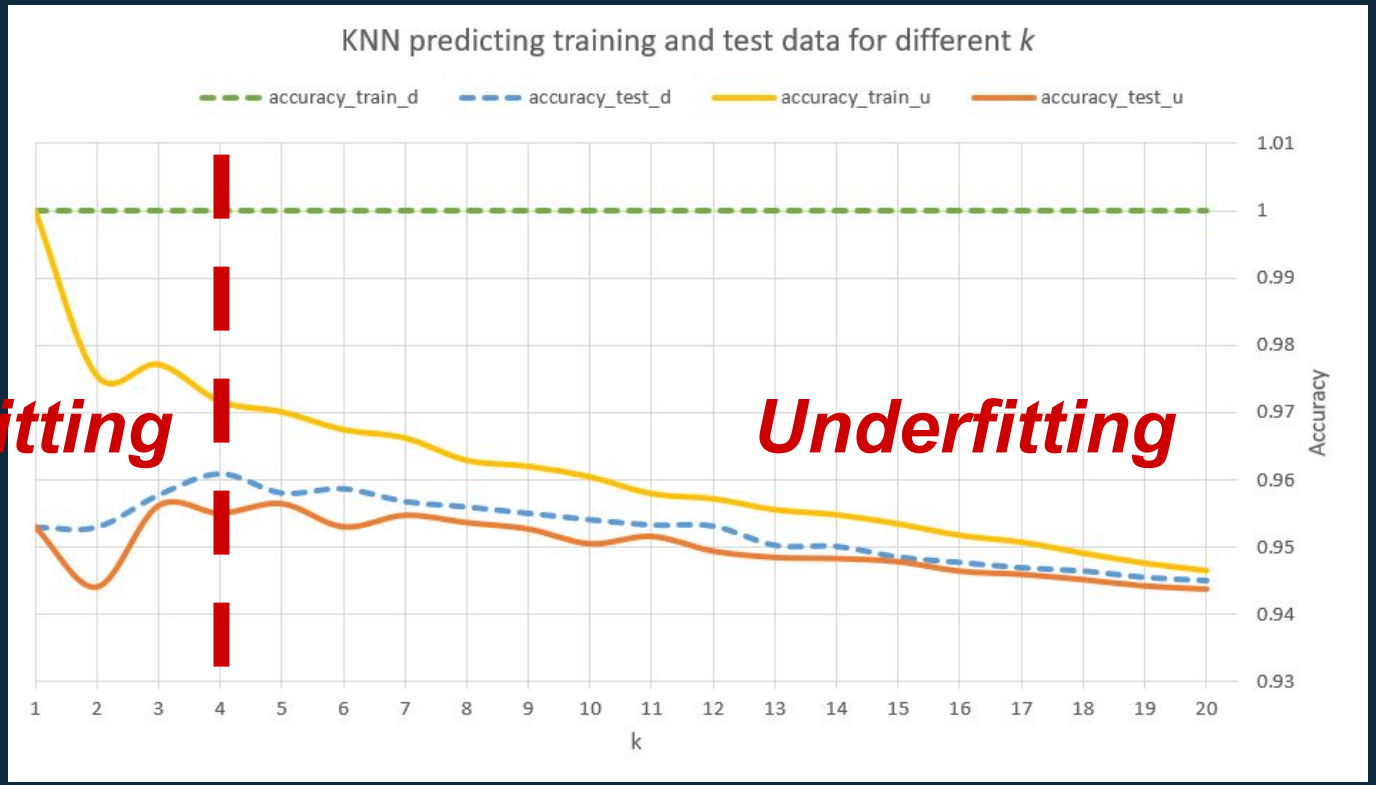
k *weights neighbors*

1. *`distance`*
2. *`uniform`*

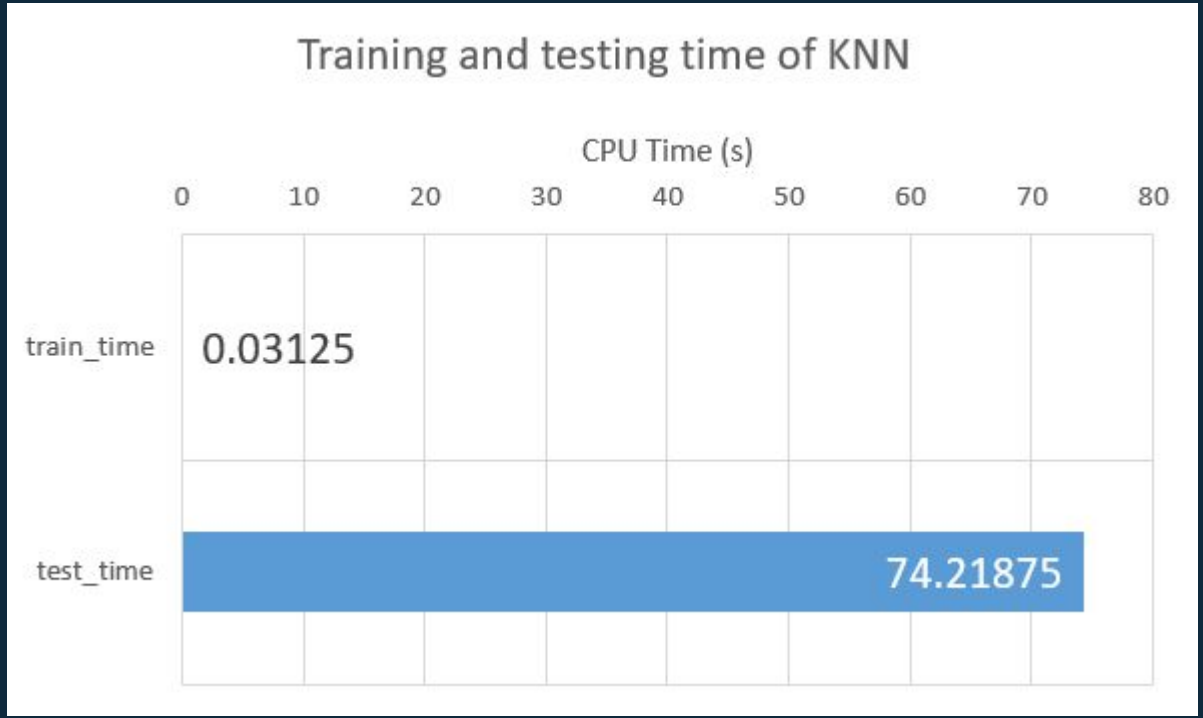
KNN


KNN for different k and weighting
pred_cpu_time_u
pred_cpu_time_d
accuracy_d
accuracy_u
CPU time (s)
Accuracy
k
62
61
60
59
58
57
56
55
54
0.97
0.968
0.966
0.964
0.962
0.96
0.958
0.956
0.954
0.952
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

KNN


KNN predicting training and test data for different k
accuracy_train_d
accuracy_test_d
accuracy_train_u
accuracy_test_u
Overfitting
Underfitting
1.01
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
Accuracy
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
k

KNN


Training and testing time of KNN
CPU Time (s)
0
10
20
30
40
50
60
70
80
train_time
0.03125
test_time
74.21875

SVM

linear

$$K(x_i, x_j) = x_i^T x_j$$

poly

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, d > 1$$

RBF

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

sigmoid

$$k(x, y) = \tanh(\alpha x^T y + c)$$

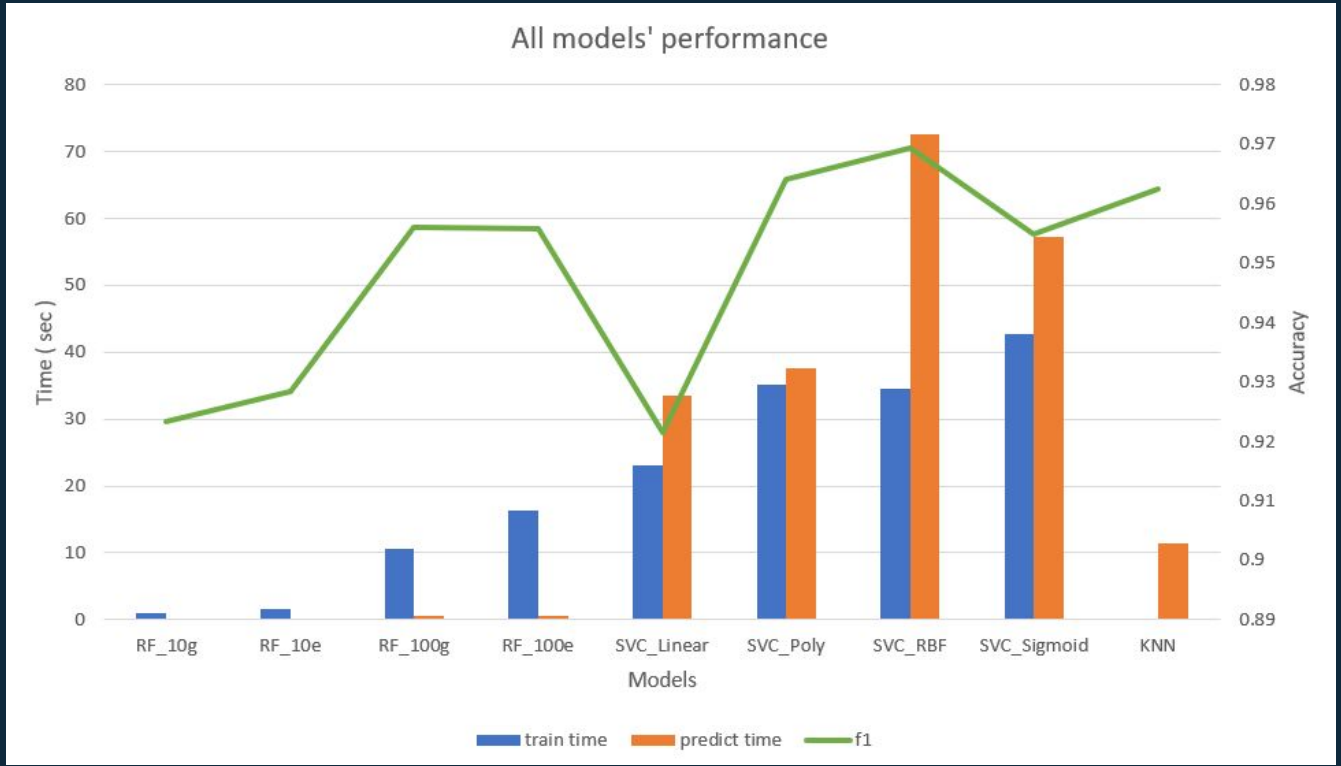
kernel – pros and cons

	pros	cons
linear	fast, simple	data must be linear-separable
poly	custom parameters	too much custom parameters
RBF	high prediction rate, less parameter	slow, tend to overfitting
sigmoid	none	hard to find proper parameters

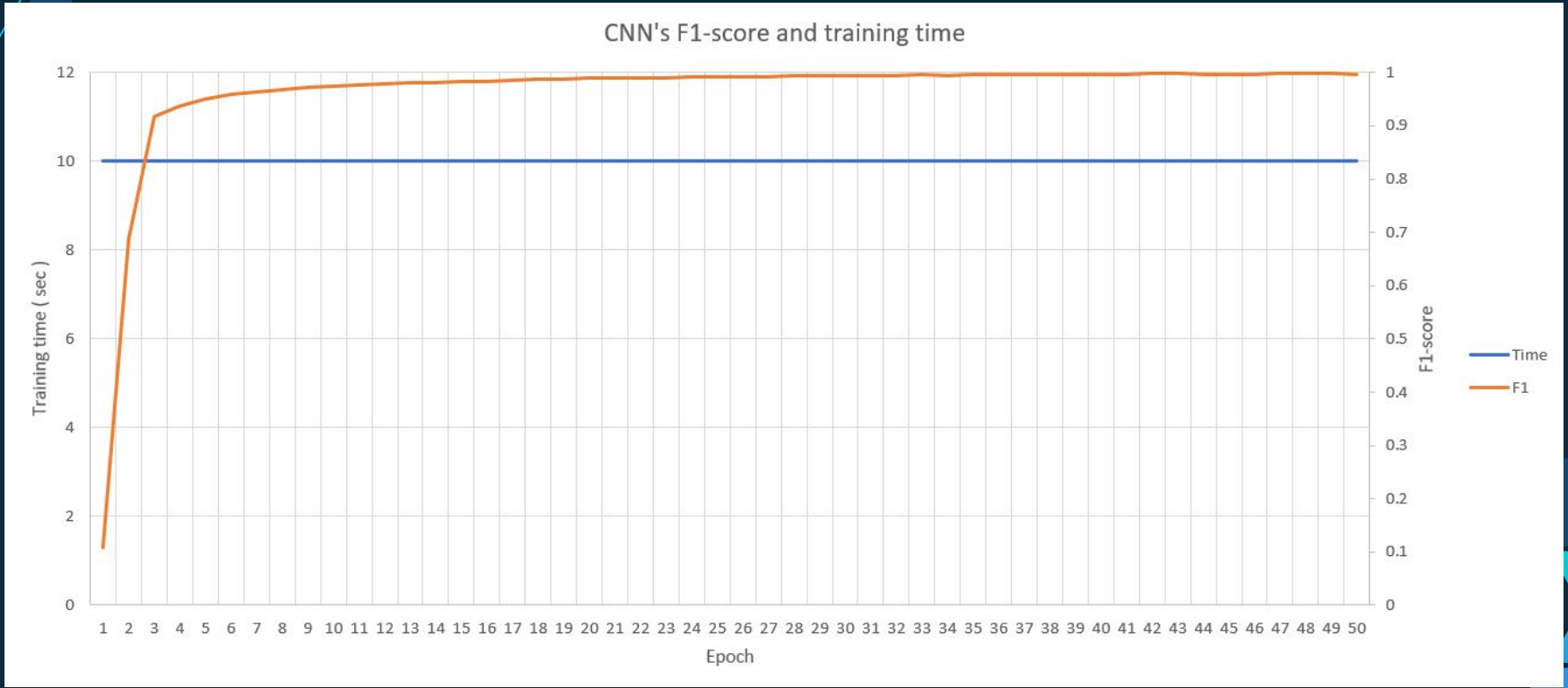
6

Result & Conclusion

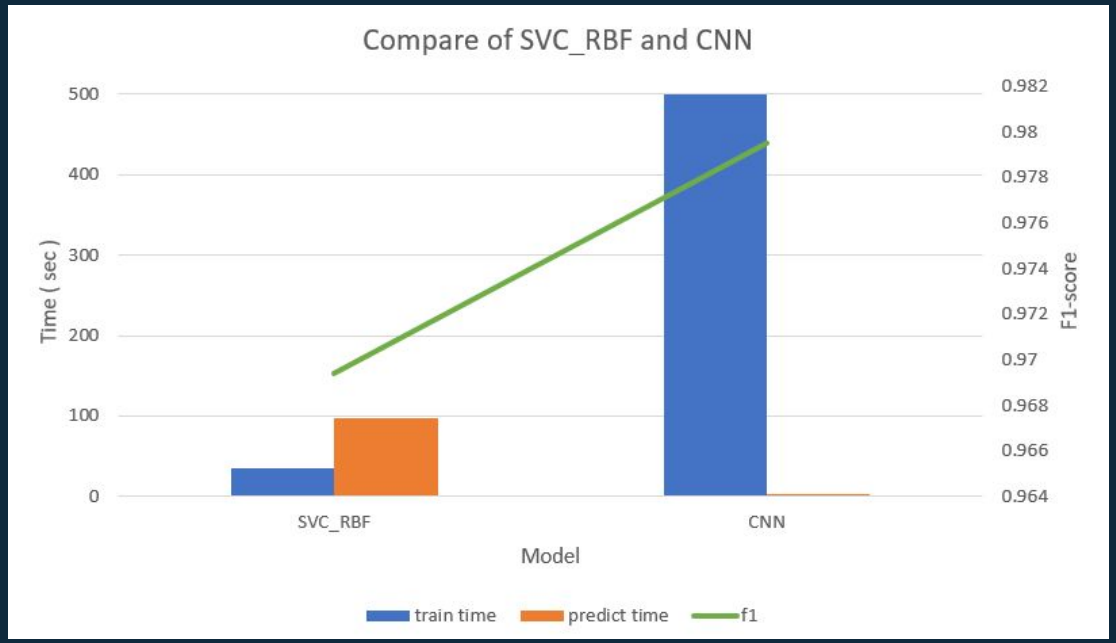
Result


All models' performance
Time (sec)
Accuracy
80
70
60
50
40
30
20
10
0
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
0.9
0.89
RF_10g
RF_10e
RF_100g
RF_100e
SVC_Linear
SVC_Poly
SVC_RBF
SVC_Sigmoid
KNN
Models
train time
predict time
f1

Conclusion


CNN's F1-score and training time
Training time (sec)
F1-score
Epoch
Time
F1
0
2
4
6
8
10
12
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

Conclusion



Convolution: 3 times Activation: Relu

Thanks!

Any questions?

Semantic Understanding Based on Lip-reading

Introduction to Data Science. Mid-Term Presentation.
2021/11/09

Team Members of Group 8

洪裕翔

資訊工程學系三年級

張金騏

資電院學士班三年級

蘇志東

資電院學士班三年級

劉子雍

資訊工程學系三年級

1 Introduction

Background Knowledge. Motivation. Project Goal (Expectation)

Background Knowledge

Lip-reading is the method of understanding speech by visually interpreting the face, lips, and tongue.

Background Knowledge

The difficulties of lip-reading:

- There are some obstructions on the speaker's face (beard, mask)
- Many people speaking at the same time
- The speaker moving while he/she is speaking

Motivation

Reducing the cost of communication and making it easier to understand what the speaker is talking about.

Project Goal (Expectation)

> The accuracy of our model can be higher than the human did (12.4%).

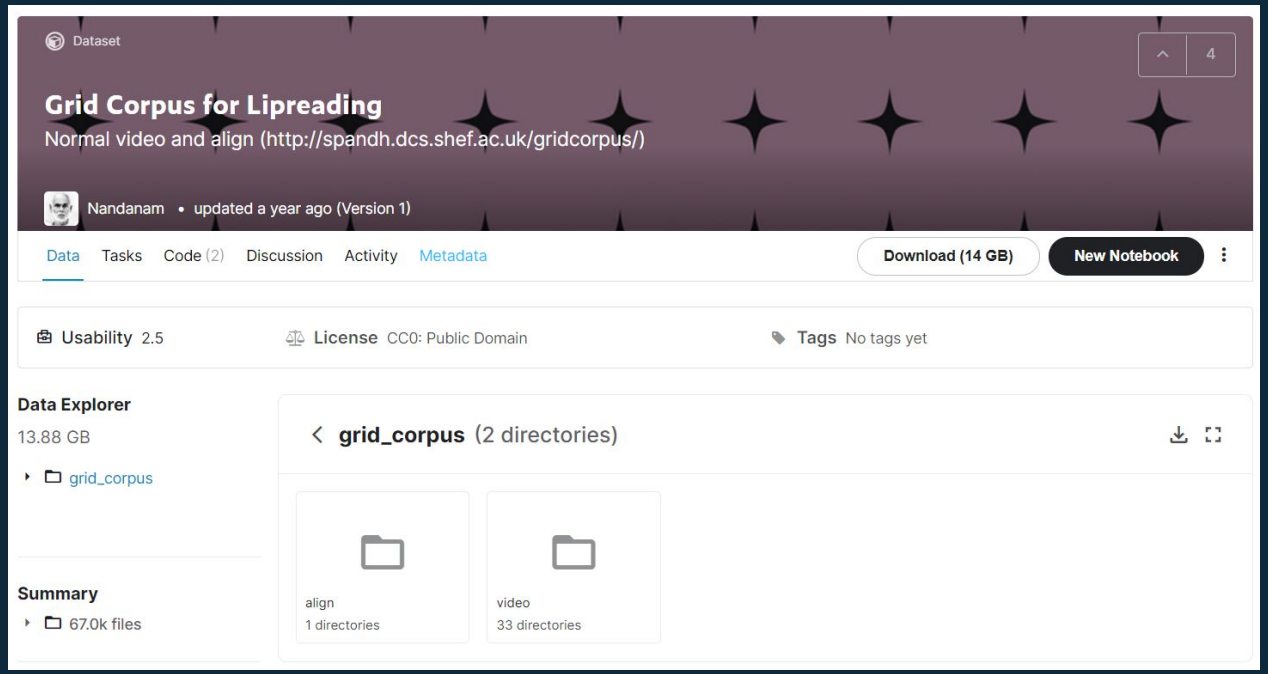
Reference: Lip Reading Sentences in the Wild [https://arxiv.org/pdf/1611.05358v1.pdf]

2

Dataset

Data Source. I/O Data

Dataset Source



Link: Grid Corpus for Lipreading | Kaggle
[https://www.kaggle.com/nandanam/grid-corpus-lipreading?select=grid_corpus]

Input

Output

```
0 23750 sil
23750 29500 bin
29500 34000 blue
34000 35500 at
35500 41000 f
41000 47250 two
47250 53000 now
53000 74500 sil
```

3 Current Progress

Current Progress

Method 1

We may make model extract features using CNN and generating **words** using LSTM directly (a.k.a. 硬train一發).

Method 2

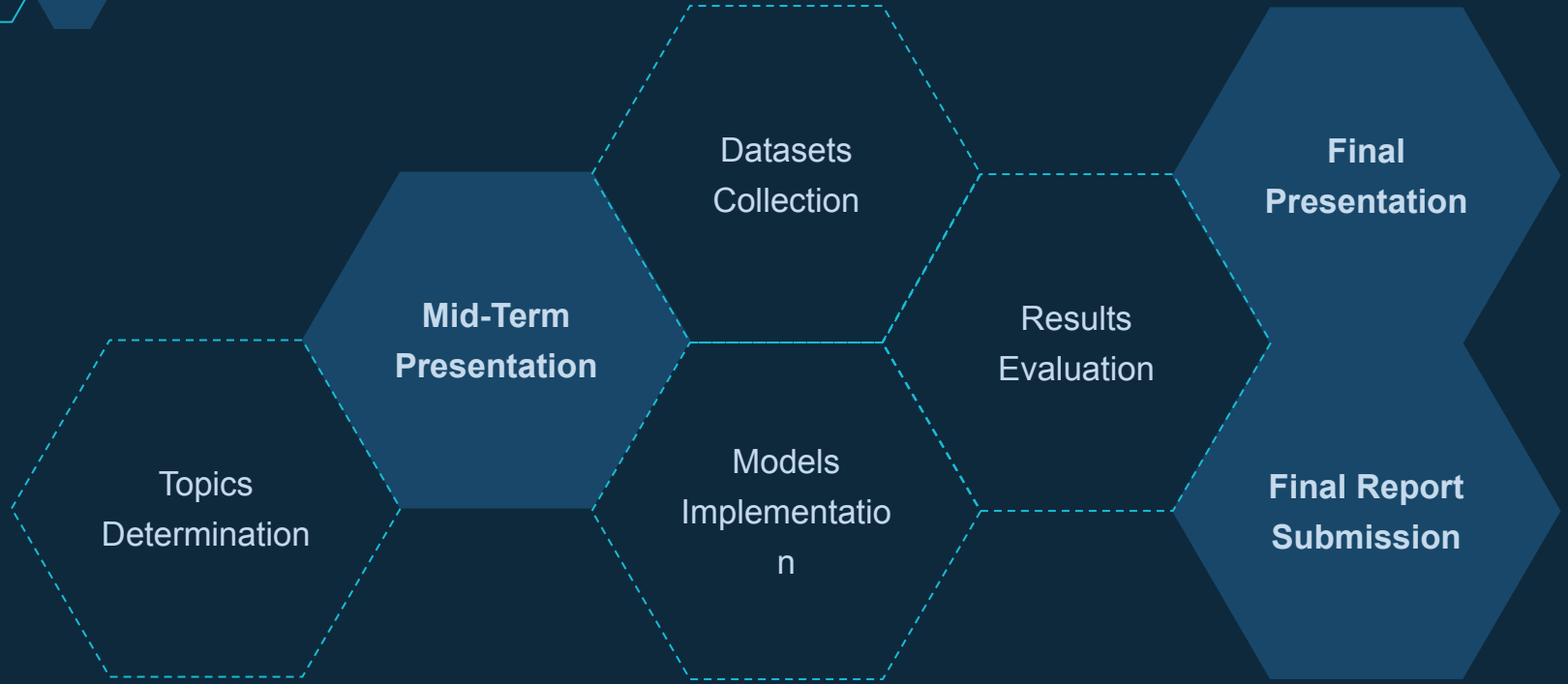
In this method, we may also use CNN and LSTM. But especially, we want the model to generate **phonics**. Then, put those phonics into a ready-made speech engine.

4

Project Schedule

Tasks to Do. Gantt Chart

Tasks to Do


Topics Determination
Mid-Term Presentation
Datasets Collection
Models Implementation
Results Evaluation
Final Presentation
Final Report Submission

Gantt Chart

	November				December				January		
	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
Topics Determination											
Mid-Term Presentation		◆									
Datasets Collection											
Models Implementation											
Results Evaluation											
Final Presentation										◆	
Final Report Submission											◆

Thanks!

Any questions?