알파벳으로 덮어씌워진 숫자 인식을 위한 기존 모델의 정확도 분석.

이상민*, 김남기**

Accuracy analysis of existing models for numerical recognition covered with alphabets.

Lee Sangmin*, and kim namgi**

요 약

Image Classification on MNIST의 정확도가 Top Accuracy 99.870를 달성한 모델의 성능을 보고 사람이 작성한 숫자를 분류해내는 정확도가 ILSVRC 대회 역대 우승 알고리즘들과 인식 에러율에서 사람의 숫자 분류 에러율을 5%로 보았을 때 사람보다 약 4.8% 더 높은 정확도를 보이는 것을 알 수 있었다. 앞서 설명한 숫자 분류의 문제는 단순히 필기체를 인식해 분류해내는 것에서 그쳤지만, DACON의 컴퓨터 비전 학습 경진 대회에서는 알파벳으로 가린 숫자의 일부분의 영역을 보아 감추어진 숫자를 예측하는 단순히 인식하여 예측하는 문제보다 더 Task가 높다고 볼 수 있다. 때문에 해당 문제에서도 기존 모델들의 정확도가 어떠할지 분석하기 위한 과정이다.

Abstract

Based on the performance of the model that achieved Top Accuracy 99.870, Image Classification on MNIST showed an accuracy of about 4.8% higher than that of ILSVRC competition winning algorithms and recognition error rates of 5%. The problem with numerical classification described earlier was simply recognizing and classifying cursive writing, but DACON's computer vision learning competition has a higher task than simply recognizing and predicting hidden numbers by looking at the areas of the numbers covered in alphabets. Therefore, it is a process to analyze the accuracy of existing models in that problem.

Key words

Overwritten number, convolution neural network, Image Classification.

Ⅰ. 서 론

* 경기대학교 소프트웨어경영대학 AI컴퓨터공학부
** 경기대학교 소프트웨어경영대학 AI컴퓨터공학부

※ 지원기관표기

Image Classification on MNIST의 정확도가 Top Accuracy 99.870를 달성한 모델의 성능을 보고 사람이 작성한 숫자를 분류해내는 정확도가 ILSVRC 대회 역대 우승 알고리즘들과 인식 에러율에서 사람의 숫자 분류 에러율을 5%로 보았을 때 사람보다 약 4.8% 더 높은 정확도를 보이는 것을 알 수 있었다. 앞서 설명한 숫자 분류의 문제는 단순히 필기체를 인식해 분류해내는 것에서 그쳤지만, DACON의 컴퓨터 비전 학습 경진 대회에서는 알파벳으로 가린 숫자의 일부분의 영역을 보아 감추어진 숫자를 예측하는 단순히 인식하여 예측하는 문제보다 더 Task가 높다고 볼 수 있다. 때문에 해당 문제에서도 기존 모델들의 정확도가 어떠할지 분석하기 위한 과정이다.

Ⅱ. DACON 컴퓨터 비전 학습 경진 대회

1. 문제에 대한 설명: Digit과 Letter를 합쳐서 만들어진 이미지에서 Letter의 범위을 넘어서는 Digit의 부분의 Pixel의 값을 0으로 낮추어 원본 숫자 이미지의 영역을 감추어 제일 오른쪽의 이미지로 만들어진 데이터를 학습 데이터로 사용합니다. 연두색의 영역은 겹쳐진 숫자의 영역이며, 연녹색의 영역은 Letter의 영역이며 해당 부분에는 감추어진 숫자가 없다는 것을 의미합니다. 이러한 색으로 나타낸 것은 시각화를 하여 편하게 보기위한 임의의 색을 채워둔 것으로 실제로는 grayscale의 이미지가 사용됩니다.

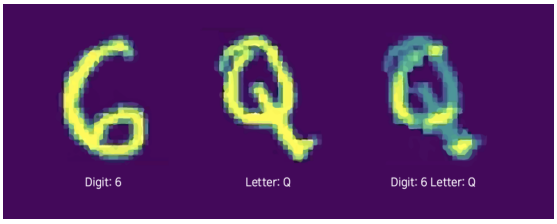
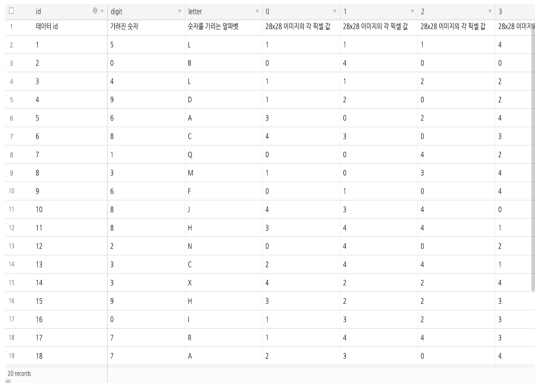


그림 2

2. 대회에서 제공되는 데이터셋
id : 데이터 id.
digit : 가려진 숫자, [0 1 2 3 4 5 6 7 8 9]

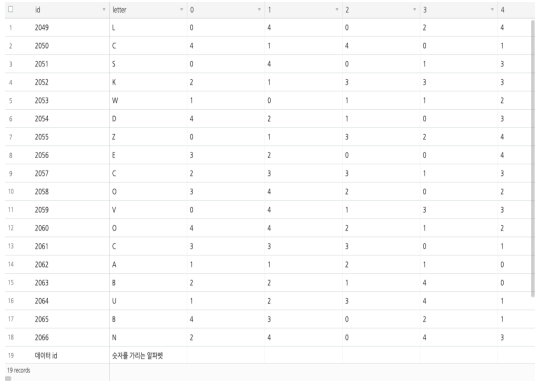
letter : 숫자를 가리는 알파벳, ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N' 'O' 'P' 'Q' 'R' 'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z']
0~784 : 28 by 28 image pixel values.



	id	digit	letter	0	1	2	3
1	데이터 id	가려진 숫자	숫자를 가리는 알파벳	28x28 이미지의 각 픽셀 값	28x28 이미지의 각 픽셀 값	28x28 이미지의 각 픽셀 값	28x28 이미지의
2	1	5	L	1	1	1	4
3	2	0	B	0	4	0	0
4	3	4	L	1	1	2	2
5	4	9	D	1	2	0	2
6	5	6	A	3	0	2	4
7	6	8	C	4	3	0	3
8	7	1	Q	0	0	4	2
9	8	3	M	1	0	3	4
10	9	6	F	0	1	0	4
11	10	8	J	4	3	4	0
12	11	8	H	3	4	4	1
13	12	2	N	0	4	0	2
14	13	3	C	2	4	4	1
15	14	3	X	4	2	2	4
16	15	9	H	3	2	2	3
17	16	0	I	1	3	2	3
18	17	7	R	1	4	4	3
19	18	7	A	2	3	0	4

20 records

그림 3 train_sample (1 ~ 2048)



	id	letter	0	1	2	3	4
1	2049	L	0	4	0	2	4
2	2050	C	4	1	4	0	1
3	2051	S	0	4	0	1	3
4	2052	K	2	1	3	3	3
5	2053	W	1	0	1	1	2
6	2054	D	4	2	1	0	3
7	2055	Z	0	1	3	2	4
8	2056	E	3	2	0	0	4
9	2057	C	2	3	3	1	3
10	2058	O	3	4	2	0	2
11	2059	V	0	4	1	3	3
12	2060	O	4	4	2	1	2
13	2061	C	3	3	3	0	1
14	2062	A	1	1	2	1	0
15	2063	B	2	2	1	4	0
16	2064	U	1	2	3	4	1
17	2065	B	4	3	0	2	1
18	2066	N	2	4	0	4	3
19	데이터 id	숫자를 가리는 알파벳					

19 records

그림 4 test_sample (2049~22528)

3. Data visualization
3.1 사람이 쉽게 예측할 수 있는 train data 예시

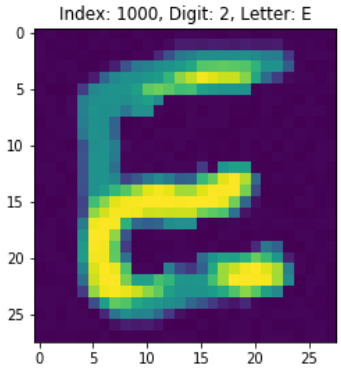


그림5 대문자 알파벳 E에 숫자 2

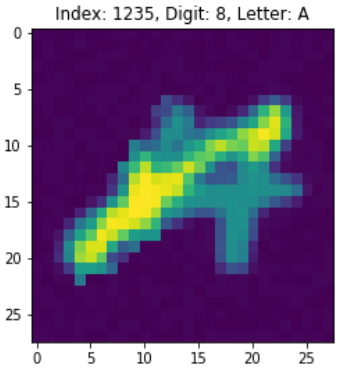


그림 8 대문자 알파벳 A에 숫자 8

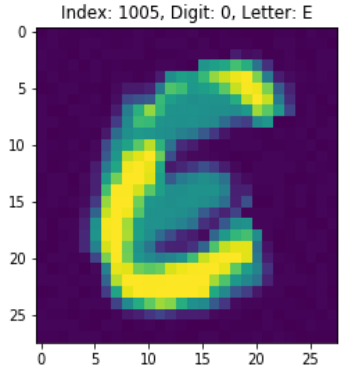


그림 6 소문자 알파벳 e에 숫자 0

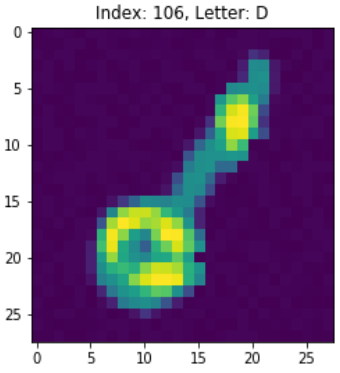


그림 9

3.2 사람이 예측하기 어려운 train data 예시.

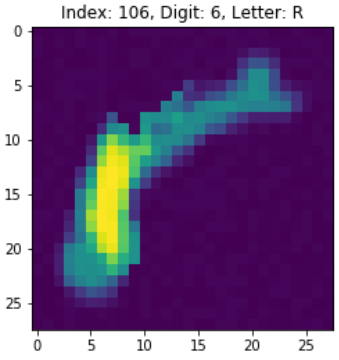


그림 7 소문자 알파벳 r에 숫자 6

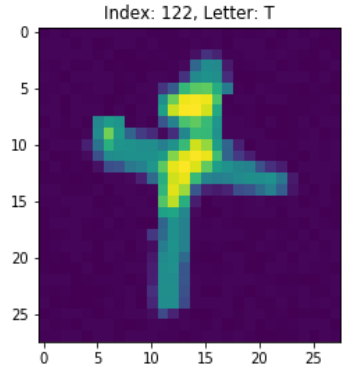
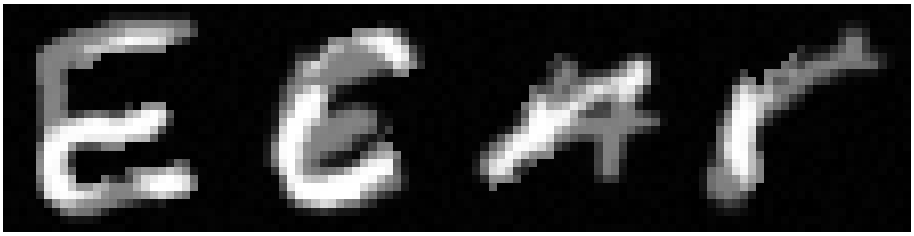


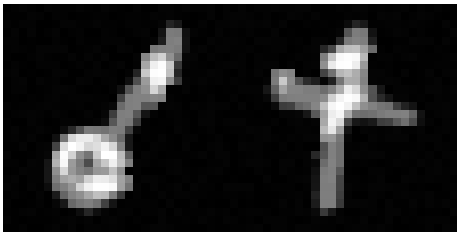
그림 10

3.3 정확도 측정을 위해 주어지는 test data set

3.4 실제 실험에 사용된 train, test images. train iamges.



test images.



4. EDA.

4.1 Data statistics.

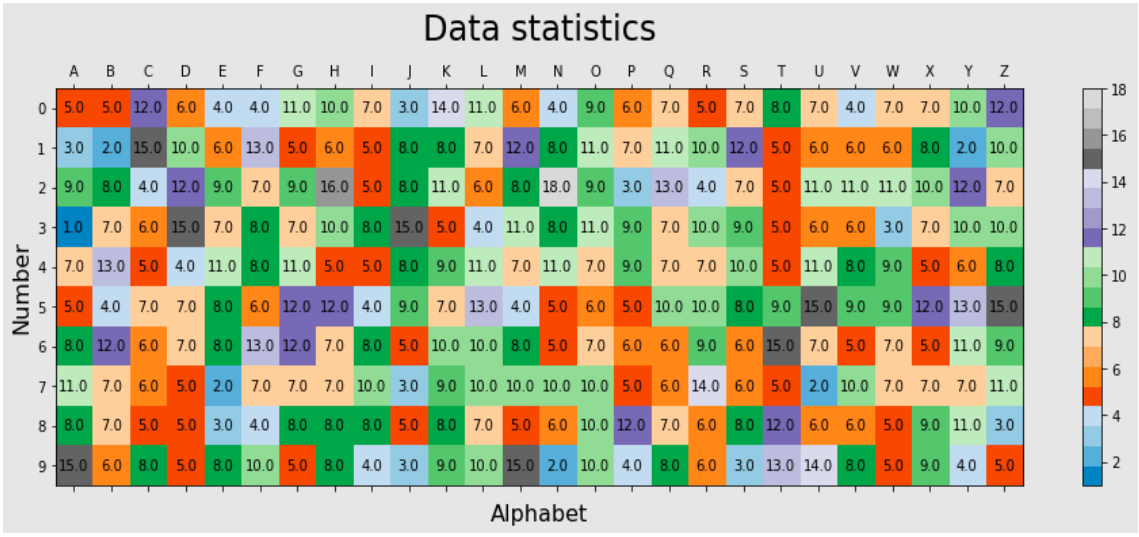


그림 16

4.2 Digit By Distribution.

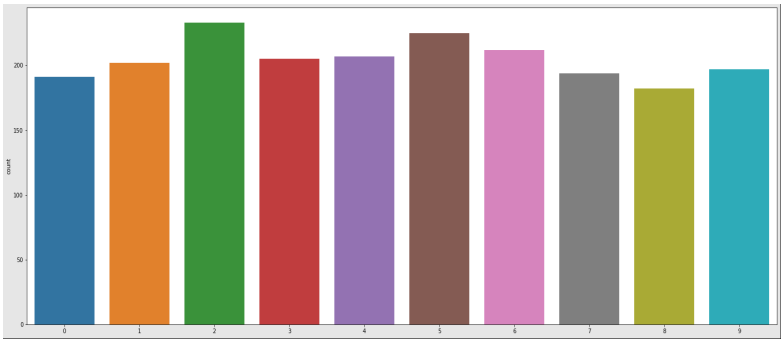


그림 18
4.3 Alphabet distribution by number.

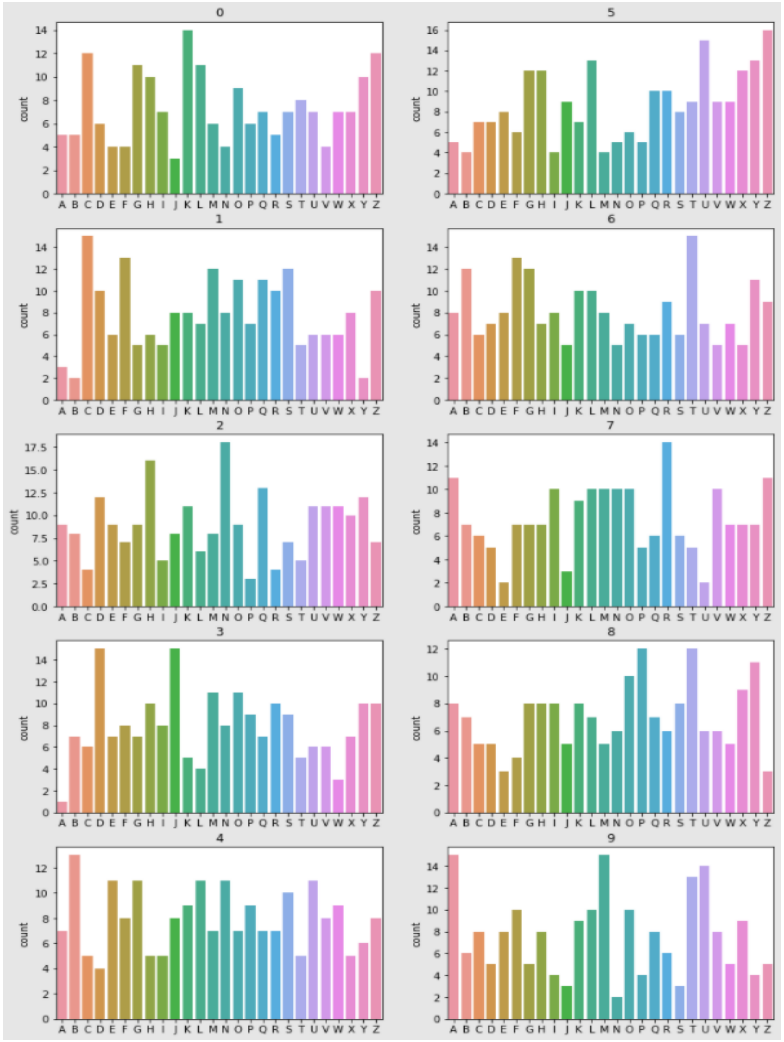


그림 19

4.4 Numeric distribution by alphabet.

그림 20

Ⅲ. 제안연구 : Convolution neural network model

VGG16, VGG19
ResNet50, ResNet101, ResNet152
ResNet50V2, ResNEt101V2, ResNet152V2
InceptionV3, InceptionResNetV2
Xception
EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3, EfficientNetB4, EfficientNetB5, EfficientNetB6, EfficientNetB7

1. 진행한 실험: Individual model Learning을 통해 정확도 추정.

해당 테스크에 가장 적합한 모델을 찾기 위해서 기존의 Image Classification 분야에서 널리 알려진 모델들을 학습시켜 가장 높은 정확도를 보인는 모델을 기준으로 세부 Parameter를 조정하여 더 정확도를 높이거나, 새로운 방식을 통해 학습을 진행하여 정확도를 높일 예정이다.

2.1 실험 진행한 환경

Google Colaboratory
CPU : Intel(R) Xeon(R) CPU @ 2.00GHz.
GPU : Tesla P100, T4, T8.
RAM : 26696424 kB.
OS : Ubuntu 18.04.5 LTS.
6.2 Data Argmentation - ImageDataGenerator

2.2 Data Argmentation's detail Parameter :
rotation_range=10, width_shift_range=0.1, height_shift_range=0.1. Random하게 생성되는 image data의 이해를 돕기위한 시각화 예시 입니다.

아래의 이미지에서 볼 수 있든 Random하게 문제의 Pixel 위치를 조정하여 Data Argmentation을 진행하였으며,

회전, 플립 등 더 다양하게 Data Argmentation을 할 수 있지만, 진행하게 될 경우 회전의 경우 6,9, 플립의 경우 모든 숫자에 사진에 나타나있는 숫자의 정보를 손실시켜 오히려 Train Data set의 혼란을 야기시켜 진행하지 않았습니다.

Train data set 2048개를 train data(1642개), validation data(406개)로 나누어서 Data Argmentation이 진행되었으며, ImageDataGenerator을 사용하여서 2048개의 이미지를 65536개로 증강하여 학습에는 Train image = 52544, Validataion image = 12992가 사용되었습니다. validation imaga는 train에 사용되지 않고 학습이 진행되었습니다.

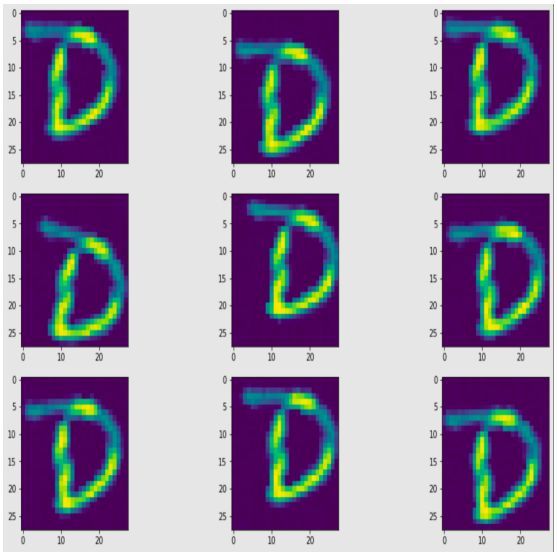


그림 21

2.3 Learning Parameter.
ImageDataGenerator (
rescale = 1./255,
validation_split = 0.2,
rotation_range = 10,
width_shift_range = 0.1,
height_shift_range = 0.1)

Batch_size = 32 (dafault)
optimizer = Adam (lr=0.002, epsilon=None)
epochs = 500

Ⅳ. 실험결과 및 분석: 초점 형상 복원을 위한

상위-1(Top-1 accuracy)과 상위-5 정확성(Top-5 accuracy)은 ImageNet의 검증 데이터셋에 대한 모델의 성능을 가리킵니다.

깊이(Depth)란 네트워크의 토폴로지 깊이를 말합니다. 이는 활성화 레이어, 배치 정규화 레이어 등을 포함합니다.

Default Input Size는 해당 모델의 최적화된 image size를 말합니다.

Input size는 모델 학습의 입력으로 사용된 이미지의 사이즈입니다.

Private score가 해당 문제의 test data의 99%를 가지고 채점된 결과이며, Public score보다 훨씬 더 높은 가치를 갖습니다.

아래의 도표에서 사용된 결과는 한번의 학습의 정확도를 기록한 것입니다.

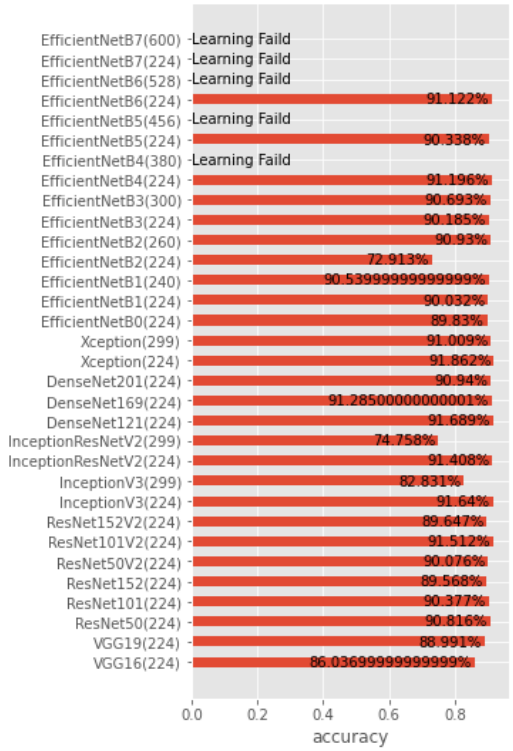


그림 22

EfficientNetB4(380x380), EfficientNetB5(456x456), EfficientNetB6(528x528), EfficientNetB7(224x224), EfficientNetB7(600x600)의 경우, Colab pro GPU memory 부족으로 인해서 학습이 불가합니다.

batch size, Layer수, Filter 갯수, input size를 줄이거나, GPU를 바꾼다 같은 방법들이 있겠지만, 위와 같이 해결하여서 학습을 진행할 경우 지금까지 진행해온 다른 모델 학습의 진행과 다르다고 판단하여 학습을 하지않았습니다.

추가적으로 EfficientNet B0-B7의 default input size의 경우 224-600의 사이즈를 가지며, 위의 작성된 default input size의 경우, 해당 모델의 최적의 input size를 작성해둔 것입니다.

VGG16, VGG19 모델을 학습하는 과정에서 학습이 진행되지 않아 원인을 찾아보니 FC layer에서 4096개에서 10개로 줄이는 과정에서 Overfitting이 발생하여서 학습이 제대로 이루어지지 않았는데 FC layer를 추가하였더니 학습이 진행되어 위와 같은 정확도를 보였지만, 다른 모델과의 학습은 추가 Layer 없이 학습이 진행되었기 때문에 같은 과정의 학습을 진행하였다고는 보기 어렵습니다.

해당 도표에서 알 수 있는 결론은 모델의 최적화된 default input size와 학습의 사용한 image의 input size가 같은 경우 가장 높은 정확도를 보인 모델은 DensetNet121 모델임을 알 수 있고, default input size와 input image size가 다른 경우의 가장 정확도가 높은 모델은 Xception임을 볼 수 있다.

InceptionResNetV2, InceptionV3, Xception 3가지 모델의 경우 default input size가 해당 모델의 최적의 input size로 알려져있는데 오히려 input size가 더 작은 경우의 정확도가 높아지는 추이를 볼 수 있으나, EfficientNetB1부터 EfficientNetB3까지의 정확도 비교를 보면 default input size를 맞추어서 학습을 진행하는 것이 더 높은 정확도를 보인다

표 1. 실험 결과 데이터 세트

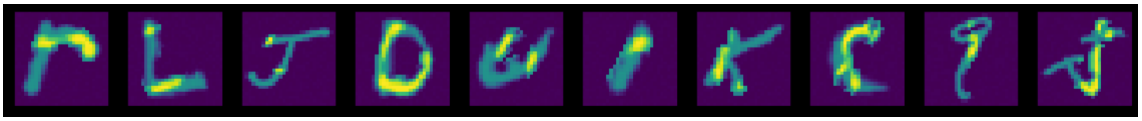
Table 1. Experimental result data sets

Public accuracy	Private accuracy	Model	Top-1 accuracy	Top-5 accuracy	Parameter	Depth	Default Input Size	Input Size
0.88235	0.86037	VGG16	0.713	0.901	138,357,544	23	224	224
0.89215	0.88991	VGG19	0.713	0.900	143,667,240	26	224	224
0.92156	0.90816	ResNet50	0.749	0.921	25,636,712	-	224	224
0.92857	0.90377	ResNet101	0.764	0.928	44,707,176	-	224	224
0.90196	0.89568	ResNet152	0.766	0.931	60,419,944		224	224
0.89215	0.90076	ResNet50V2	0.760	0.930	25,613,800		224	224
0.91666	0.91512	ResNet101V2	0.772	0.938	44,675,560		224	224
0.89705	0.89647	ResNet152V2	0.780	0.942	60,380,648		224	224
0.92156	0.91640	InceptionV3	0.779	0.937	23,851,784	159	299	224
0.81862	0.82831	InceptionV3	0.779	0.937	23,851,784	159	299	299
0.94117	0.91408	InceptionResNetV2	0.803	0.953	55,873,736	572	299	224
0.73039	0.74758	InceptionResNetV2	0.803	0.953	55,873,736	572	299	299
0.93137	0.91689	DenseNet121	0.750	0.923	8,062,504	121	224	224
0.92156	0.91285	DenseNet169	0.762	0.932	14,307,880	169	224	224
0.91666	0.90940	DenseNet201	0.773	0.936	20,242,984	201	224	224
0.94117	0.91862	Xception	0.790	0.945	22,910,480	126	299	224
0.93137	0.91009	Xception	0.790	0.945	22,910,480	126	299	299
0.91666	0.89830	EfficientNetB0	0.763	0.932	5.3M	-	224	224
0.90686	0.90032	EfficientNetB1	0.788	0.944	7.8M	-	240	224
0.92647	0.90540	EfficientNetB1	0.788	0.944	7.8M	-	240	240
0.74019	0.72913	EfficientNetB2	0.798	0.949	9.2M	-	260	224
0.94117	0.90930	EfficientNetB2	0.798	0.949	9.2M	-	260	260
0.92156	0.90185	EfficientNetB3	0.811	0.955	12M	-	300	224
0.92156	0.90693	EfficientNetB3	0.811	0.955	12M	-	300	300
0.91176	0.91196	EfficientNetB4	0.826	0.963	19M	-	380	224
x	x	EfficientNetB4	0.826	0.963	19M	-	380	380
0.93137	0.90338	EfficientNetB5	0.833	0.967	30M	-	456	224
x	x	EfficientNetB5	0.833	0.967	30M	-	456	456
0.95588	0.91122	EfficientNetB6	0.840	0.969	43M	-	528	224
x	x	EfficientNetB6	0.840	0.969	43M	-	528	528
x	x	EfficientNetB7	0.844	0.971	66M	-	600	224
x	x	EfficientNetB7	0.844	0.971	66M	-	600	600

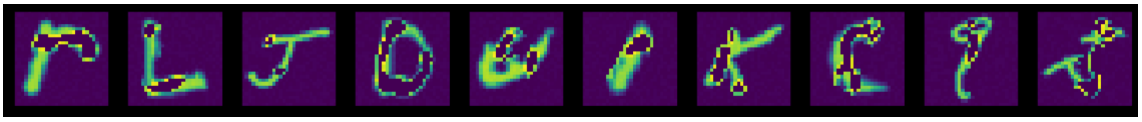
V. 결 론

1. model ensamble.
2. Validation K-fold.
3. Parameter optimization.
4. Letter 정보를 활용한 학습. (숫자가 무조건 있는 영역, 숫자가 무조건 없는 영역, 숫자가 있을 수 있는 영역)

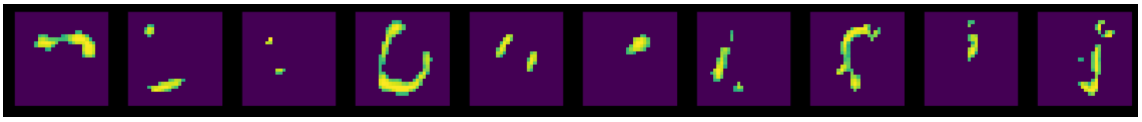
4.1 Original images



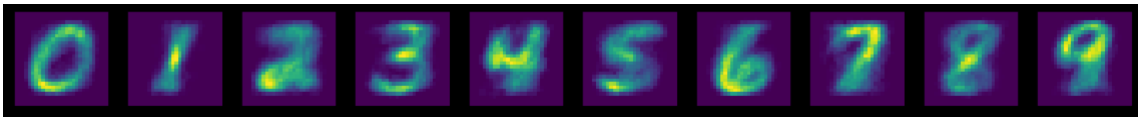
4.2 Letter part.



4.3 Digit part.



4.4 All of Digit Sum.



Origin image에서 letter 부분을 제외하고 Digit 부분만 모두 모아서 시각화한 결과 4번과 같은 이미지의 형상을 볼 수 있는데 해당 이미지로 유추해보아 Original image를 Digit 부분과 Letter 부분을 같이 학습하여 예측을 진행할 경우 더 높은 정확도를 보일 수 있을거라고 생각한다.

참 고 문 헌

[1] S. K. Nayar and Y. Nakagawa, "Shape from focus", IEEE Trans. Pattern Anal. Machine Intell., vol. 16, pp. 824-831, August 1994.

[2] H. N. Nair and C. V. Stewart, "Robust focus ranging", Proc. CVPR, pp. 309-314, 1992.

[3] M. Subbarao and T. S. Choi, "Accurate recovery of three-dimensional shape from image focus", IEEE Trans. Pattern Anal. Machine Intell., vol. 17, pp. 266-274, March 1995.

[4] T. S. Choi and J. Yun, "Three-dimensional shape recovery from focused image surface". Opt. Eng., vol. 39, May 2000.

[5] M. Asif and T. S. Choi, "Shape from focus using multilayer feedforward neural network", IEEE Transaction on Image Processing, vol. 10, no. 11, pp. 1670-1675, November 2001.

한글제목	휴먼명조, 17, 장평:90, 자간: -7
저자명	돋움, 11, 장평:90, 자간: 5
영문제목	견명조, 15, 장평:90, 자간: -7
영문저자명	휴먼명조, 10, 장평:90, 자간: 5
요약본문	휴먼명조, 9.2, 장평:90, 자간:-6
영문요약문	영문:Times New Roman, 9.2, 장평:90, 자간:-6
각장제목	휴먼고딕, 11, 장평:90, 자간:-6
본문내용	휴먼명조, 10, 장평:90, 자간:-6
소제목	중고딕 11, 장평:90, 자간:-6
그림캡션	중고딕, 9, 장평:90, 자간:-6
표캡션	중고딕, 9, 장평:90, 자간:-6
식	크기 9, 왼정렬, 식번호는 오른 정렬
참고문헌	영문:Times New Roman, 10, 장평:90, 자간:-6