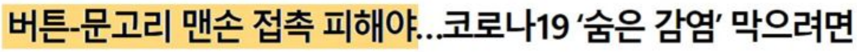
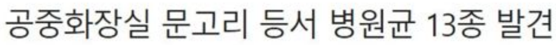
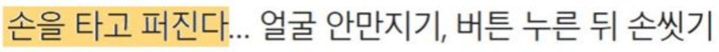
Happy Feet
: A Brand New Communication Method Using Feet Position

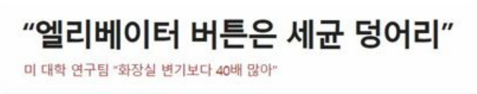
Team 16
Jio Oh, Donghyeon Lee, Jihyeong Hong

Problems in Life


버튼-문고리 맨손 접촉 피해야...코로나19 '숨은 감염' 막으려면

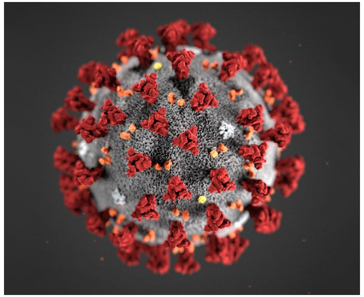

공중화장실 문고리 등서 병원균 13종 발견


손을 타고 퍼진다... 얼굴 안만지기, 버튼 누른 뒤 손씻기


"엘리베이터 버튼은 세균 덩어리"

미 대학 연구팀 "화장실 변기보다 40배 많아"

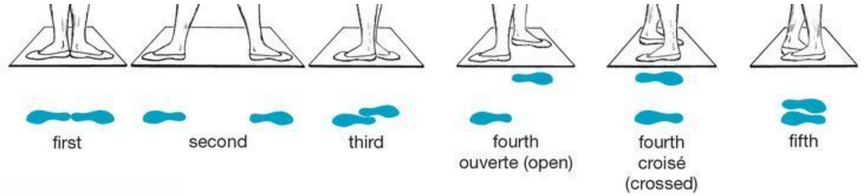
Elevator buttons have 40x more bacteria than the toilet



Motivations

#Why not?!

Data Generation

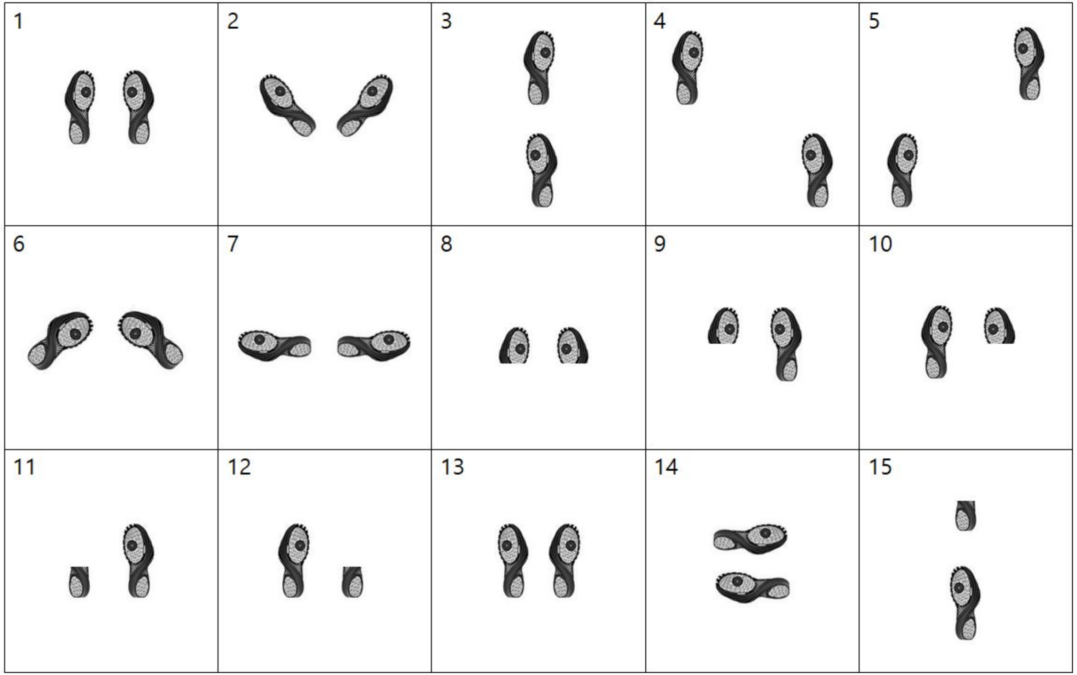


Ballet Position

Feet Position Selection

(1) Easily Done by Users
    (a) Existing feet positions like ballet position
(2) Well Classified

30 Candidates -> 15 Classes

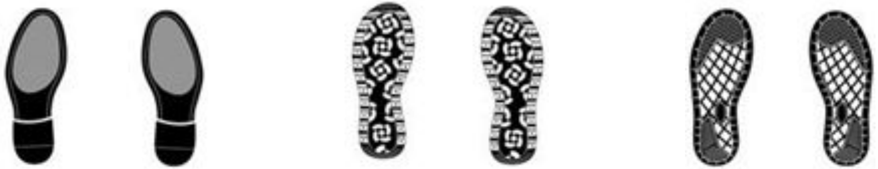


Data Generation

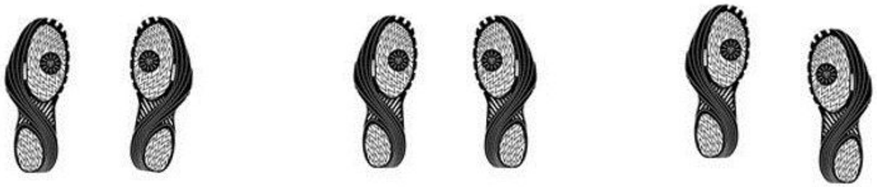
Handmade 400 variations per class

One class

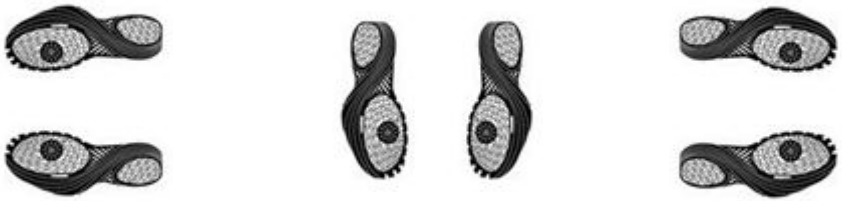
20 Different soles



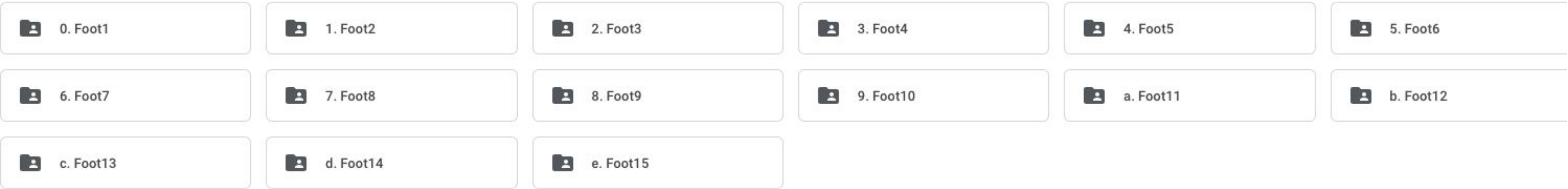
5 Different positions



4 Rotation



Input Data Overview



0. Foot1
1. Foot2
2. Foot3
3. Foot4
4. Foot5
5. Foot6
6. Foot7
7. Foot8
8. Foot9
9. Foot10
a. Foot11
b. Foot12
c. Foot13
d. Foot14
e. Foot15

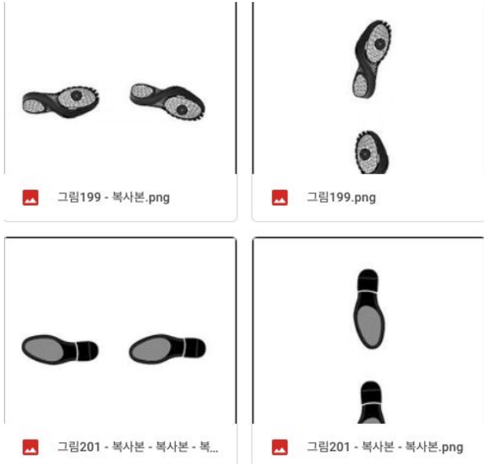


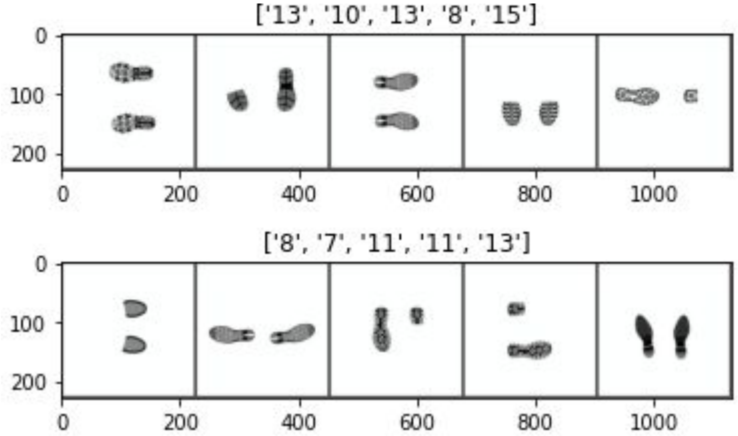
그림199 - 복사본.png

그림199.png

그림201 - 복사본 - 복사본 - 복...

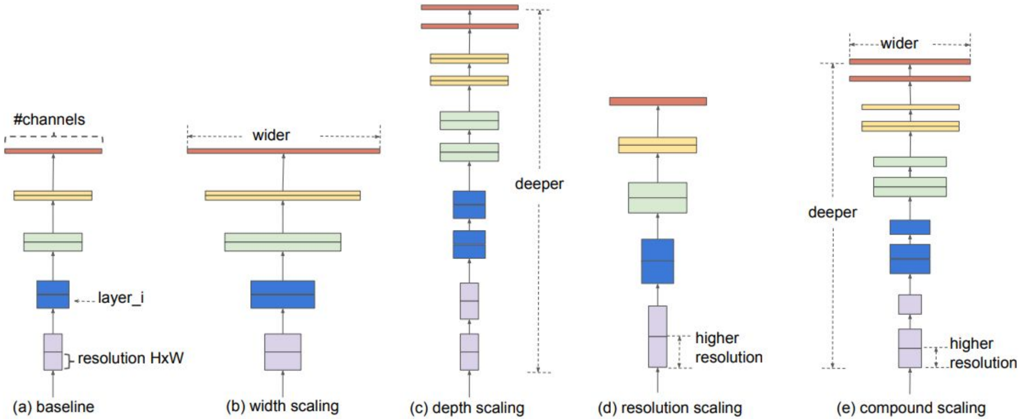
그림201 - 복사본 - 복사본.png

Classification
(DataLoader)



Model

Classification model: Efficient-Net



(a) baseline (b) width scaling (c) depth scaling (d) resolution scaling (e) compound scaling

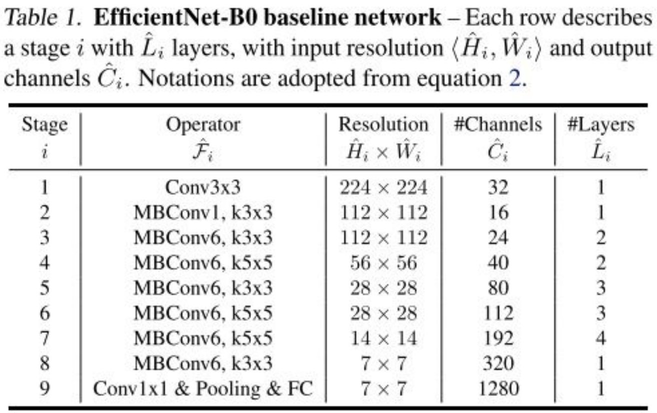
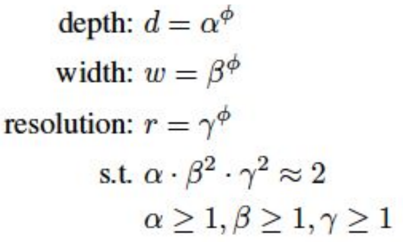
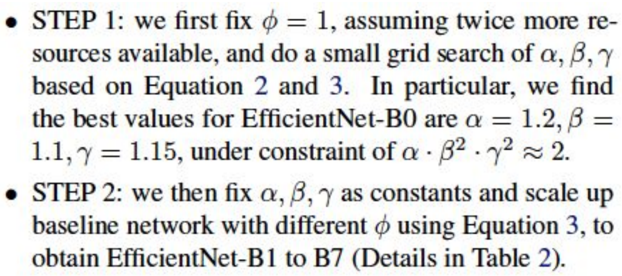


Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	28×28	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1



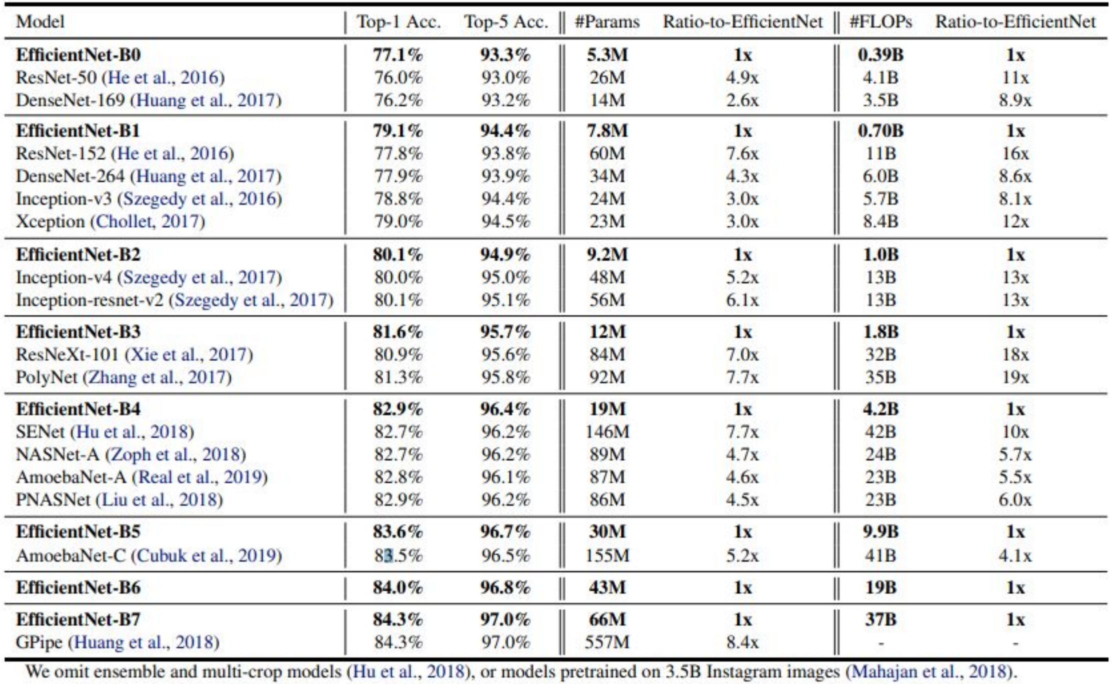
$$
\begin{aligned}
\text{depth: } & d = \alpha^\phi \\
\text{width: } & w = \beta^\phi \\
\text{resolution: } & r = \gamma^\phi \\
\text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\
& \alpha \geq 1, \beta \geq 1, \gamma \geq 1
\end{aligned}
$$



- STEP 1: we first fix $\phi = 1$, assuming twice more resources available, and do a small grid search of α, β, γ based on Equation 2 and 3. In particular, we find the best values for EfficientNet-B0 are $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$, under constraint of $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$.
- STEP 2: we then fix α, β, γ as constants and scale up baseline network with different ϕ using Equation 3, to obtain EfficientNet-B1 to B7 (Details in Table 2).

Model

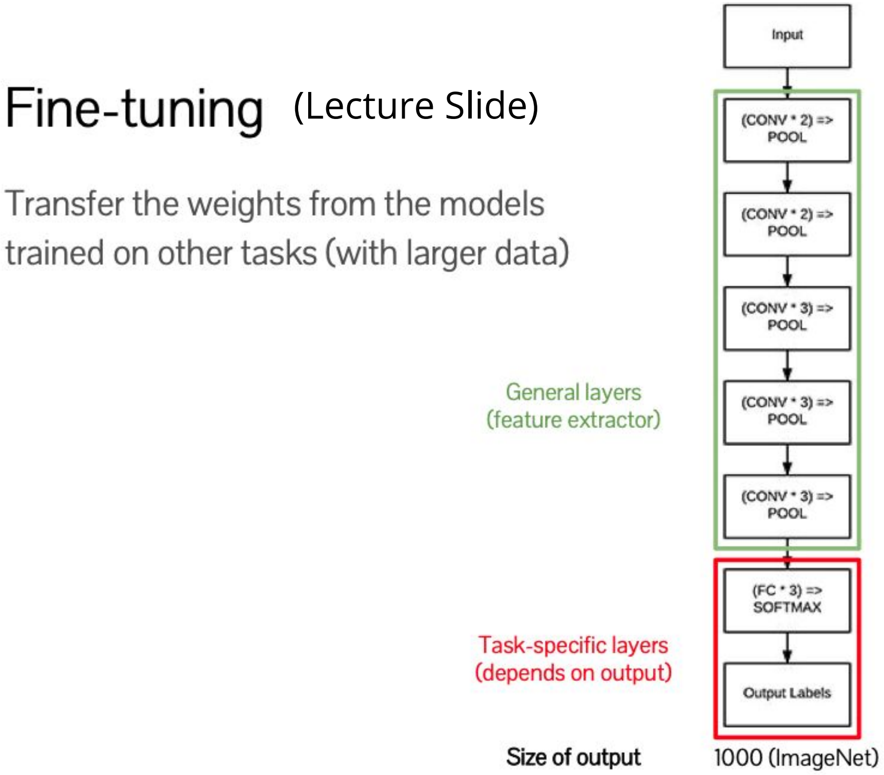
Classification model: Efficient-Net



Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	**77.1%**	**93.3%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**79.1%**	**94.4%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**80.1%**	**94.9%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.6%**	**95.7%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**82.9%**	**96.4%**	**19M**	**1x**	**4.2B**	**1x**
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.6%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.0%**	**96.8%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.3%**	**97.0%**	**66M**	**1x**	**37B**	**1x**
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

We omit ensemble and multi-crop models (Hu et al., 2018), or models pretrained on 3.5B Instagram images (Mahajan et al., 2018).

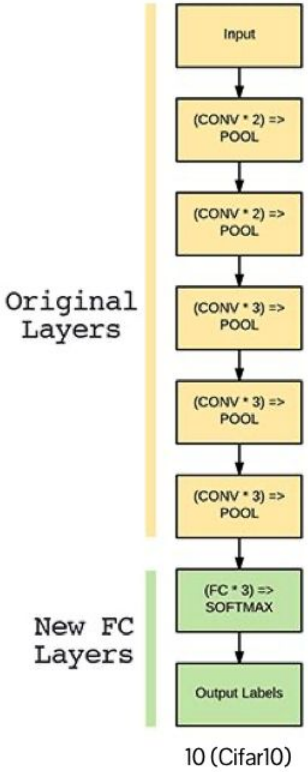
Model - Transfer Learning



Fine-tuning

(Lecture Slide)

Transfer the weights from the models trained on other tasks (with larger data)

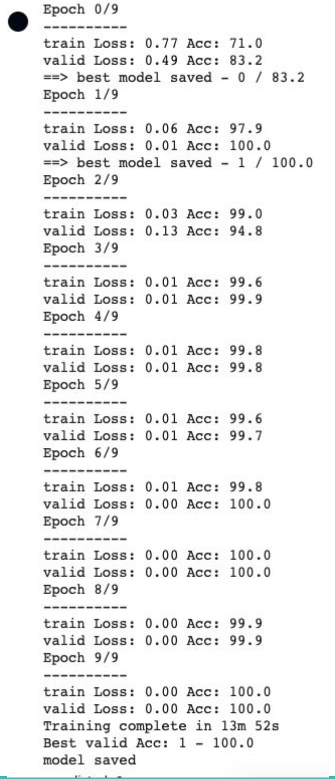


Pre-Trained
Efficient-b0 (w/ ImageNet)

Tuning with Footsteps

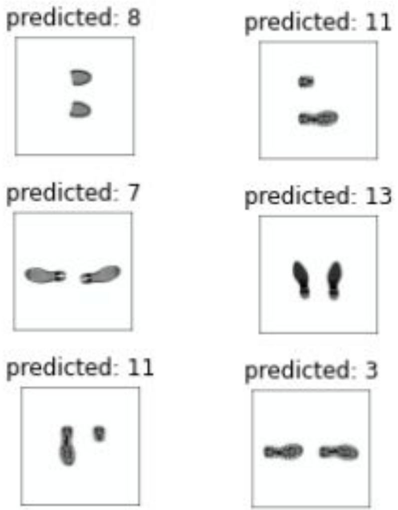
Results

Accuracy for Training / Validation Sets

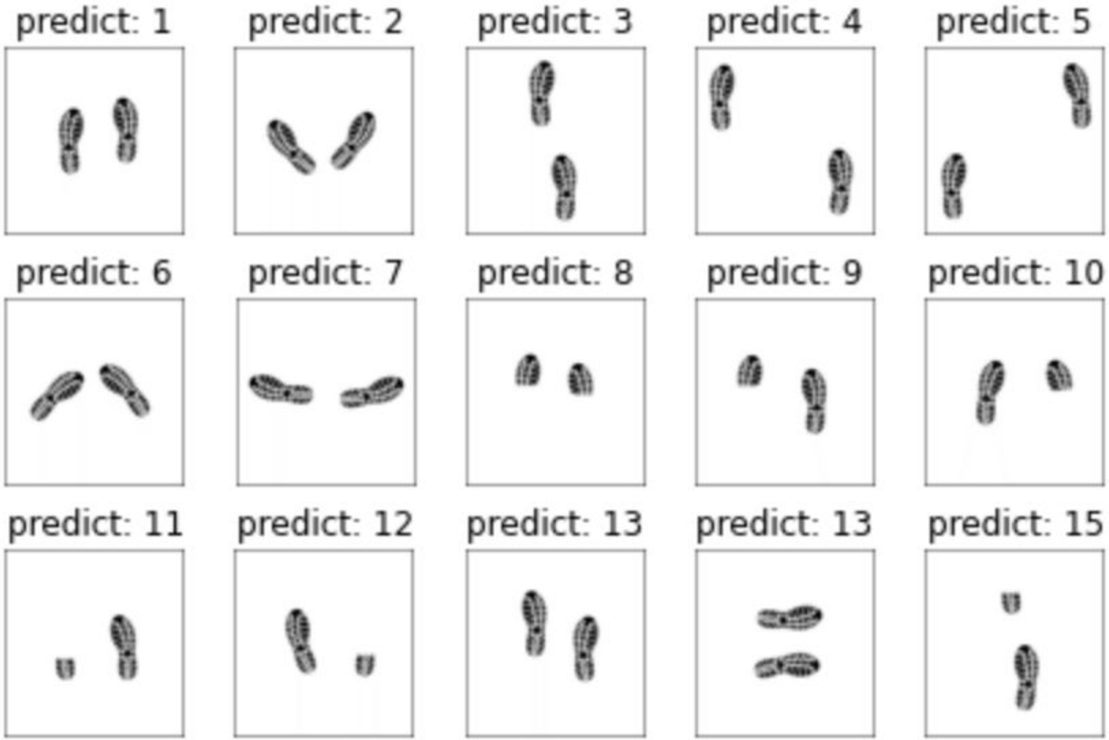


```
Epoch 0/9
----------
train Loss: 0.77 Acc: 71.0
valid Loss: 0.49 Acc: 83.2
==> best model saved - 0 / 83.2
Epoch 1/9
----------
train Loss: 0.06 Acc: 97.9
valid Loss: 0.01 Acc: 100.0
==> best model saved - 1 / 100.0
Epoch 2/9
----------
train Loss: 0.03 Acc: 99.0
valid Loss: 0.13 Acc: 94.8
Epoch 3/9
----------
train Loss: 0.01 Acc: 99.6
valid Loss: 0.01 Acc: 99.9
Epoch 4/9
----------
train Loss: 0.01 Acc: 99.8
valid Loss: 0.01 Acc: 99.8
Epoch 5/9
----------
train Loss: 0.01 Acc: 99.6
valid Loss: 0.01 Acc: 99.7
Epoch 6/9
----------
train Loss: 0.01 Acc: 99.8
valid Loss: 0.00 Acc: 100.0
Epoch 7/9
----------
train Loss: 0.00 Acc: 100.0
valid Loss: 0.00 Acc: 100.0
Epoch 8/9
----------
train Loss: 0.00 Acc: 99.9
valid Loss: 0.00 Acc: 99.9
Epoch 9/9
----------
train Loss: 0.00 Acc: 100.0
valid Loss: 0.00 Acc: 100.0
Training complete in 13m 52s
Best valid Acc: 1 - 100.0
model saved
```

Validation Set Prediction Results



Results



Prediction Results for
Newly Generated Data

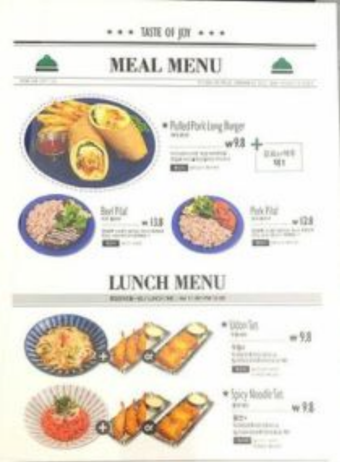
Simple Demo

Applications

- Food & Drink Orders

Drinks

Red Wine



• • • TASTE OF JOY • • •

MEAL MENU

- Pulled Pork Long Burger w9.8
- Beef Pilaf w13.8
- Pork Pilaf w12.8

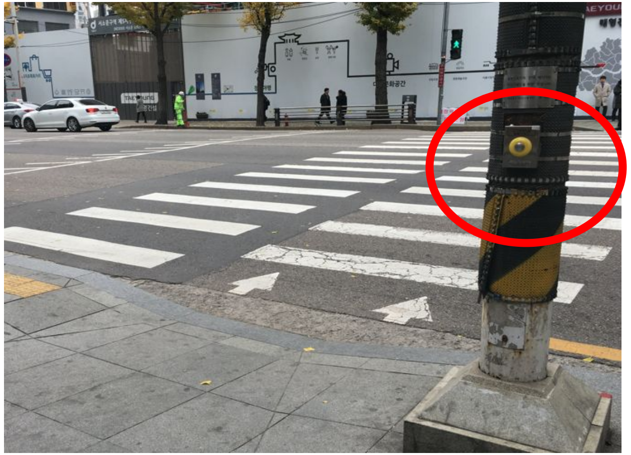
LUNCH MENU

- Udon Set w9.8
- Spicy Noodle Set w9.8

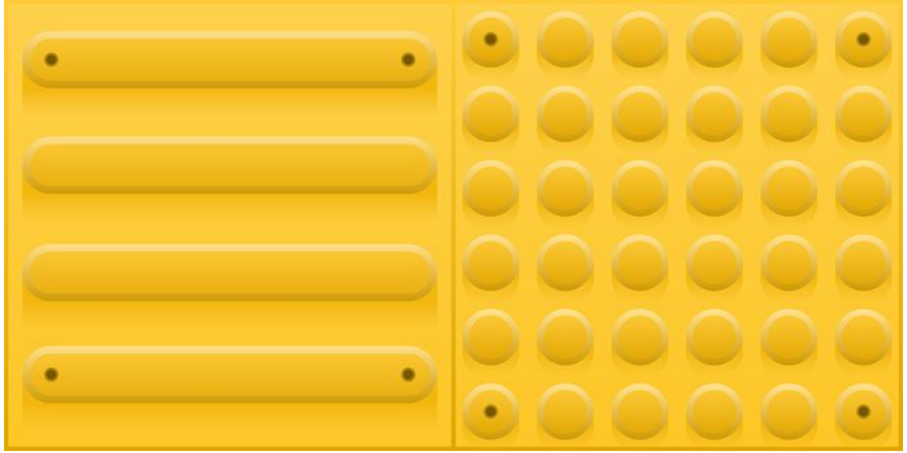
Convenient & Save Labor Cost

Applications

- Crosswalk Assistance for Blind People



Hard to Find



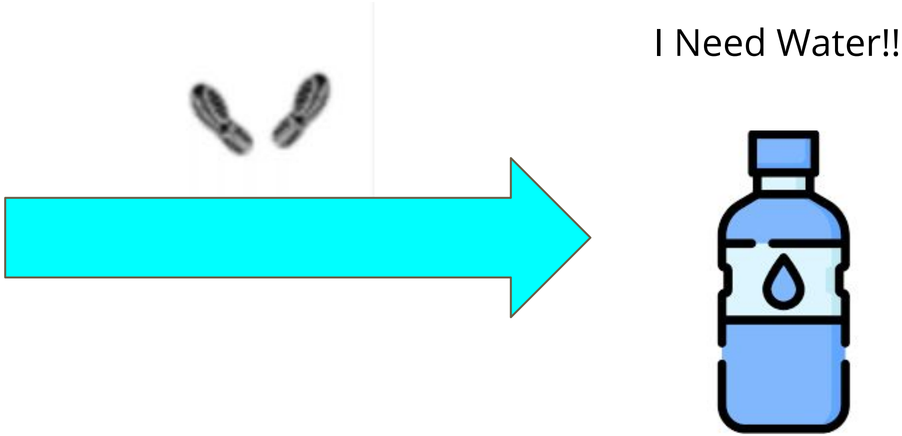
Foot Sensors on Braille Blocks

Applications

- Answering Questions in Classes
- Simple Polls in Discussions

Applications

- Simple Communications/Signals in Orchestra for Musicians
  - Mostly Out of Hands
  - Need of Water, Change of Songs/Sheets



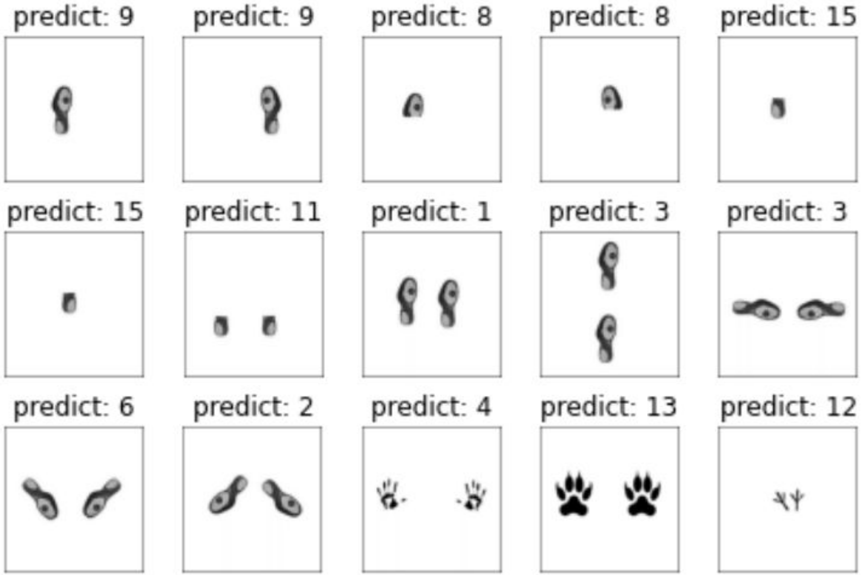
I Need Water!!

Applications

- Expansion to Games
  - Pump Games
  - VR Games

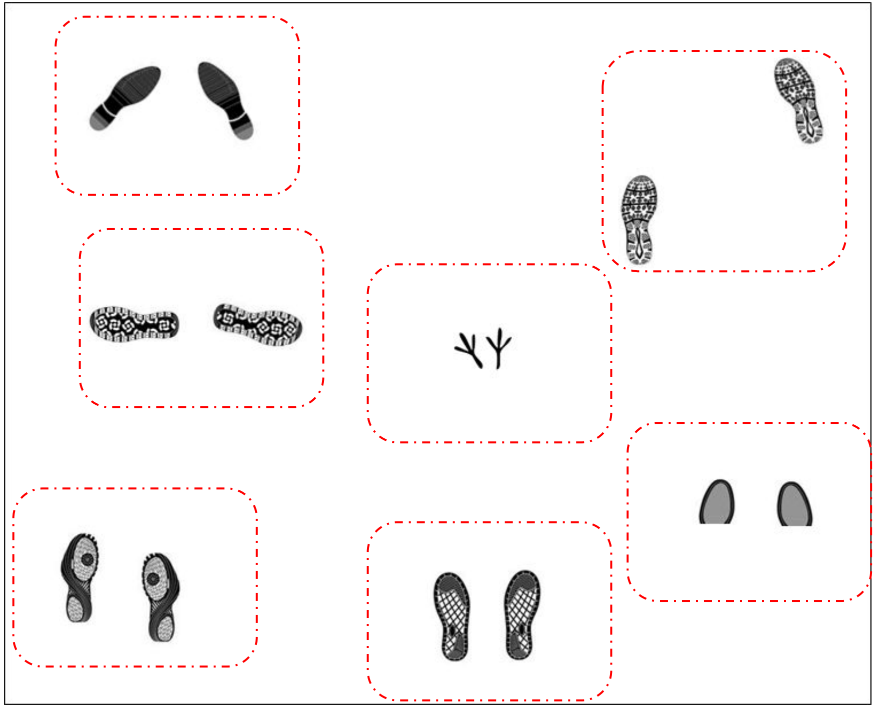
Discussions/Further Development

- Filtering Wrong Shapes
  - Currently, wrong shapes are evaluated as valid



Discussions/Further Development

- Classification of Multiple Footprints
  - Multiple Users
  - Bounding Box Regression



Thank You