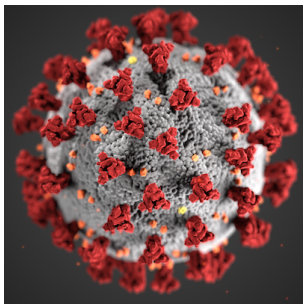
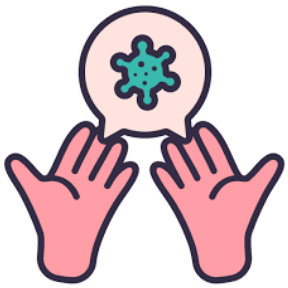
[CS470] Final Project Report: Happy Feet

Team 16
20180396 Jio Oh
20180442 Donghyeon Lee
20180731 Jihyeong Hong
https://github.com/harryoh99/CS470_team16

<Motivation>



Due to the widespread coronavirus, our lives have totally changed. We try to avoid physical contact with others and touching public facilities that are prone to have bacteria or molecules of diseases. One of the most bacteria-exposed places are elevators, in which all the residents and visitors of the building press the buttons with their probably contaminated hands. This problem is becoming a major issue nowadays and people are developing elevators that move via audio signals or attach antibiotic stickers on elevator buttons. Plus, we all went through some difficulties when pressing elevator buttons when holding heavy loads with our hands full. It is very bothersome to put our load down and pick it up again just to click the button. This problem gets even worse when the elevator is full, where we need to ask for help to press the button for us, which is one of the most challenging tasks for introverted people.



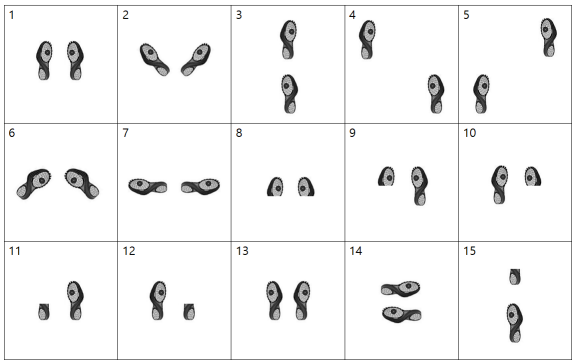
Nowadays, most communications and signal transmissions are done by hands, which is considered as the major sources of bacteria transitions. In order to solve the above problems we tried to flip the coin and had a glimpse of thought, 'Why not use feet for communication?'. Though a bit less expressive, feet are free from viruses and can be a good alternative for expressions, since we utilize it every day. Our team decided to use the photos of footsteps as the input data, which can be most easily collected in the real world.

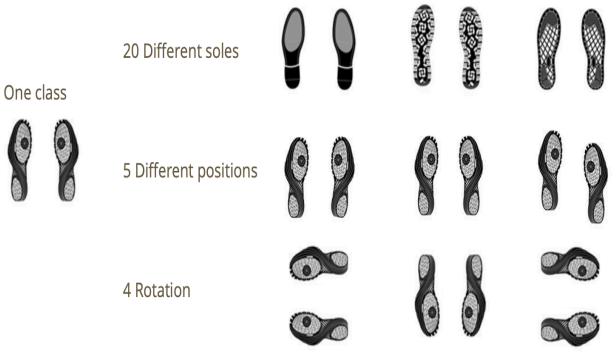
<Data Generation: Preprocessing>

We assumed that the footsteps will be collected with the help of cameras, thus we tried to collect the photos of various footsteps, with various shoe sole shapes and feet positions. These data weren't easily found on the internet, so we decided to generate the data by ourselves. 30 feet position were selected Among 30 feet position candidates, we extracted 15[1], with the following two criteria: Are the corresponding feet position can easily be done by all users and is familiar to them? Can they be well classified by the model and the

[1] **Novel Contribution**: We defined and discovered 15 well-classified, easily mimicked foot positions and constructed corresponding datasets that can be easily utilized as a basis in the future. Plus, footsteps are easily collectable, thus the idea has high scalability.

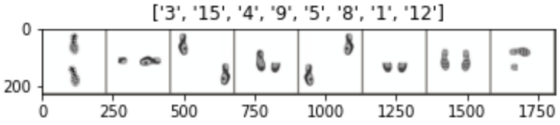
users? (The familiarity was assessed based on our usual habits and ballet feet positions and the “well-classified” part was issued in order to prevent confusion for both the model and the users.)



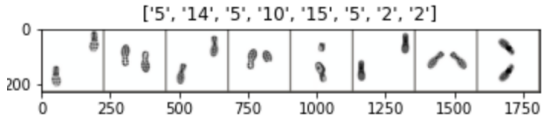


After the filtration, we got the above 15 classes, where we assigned number 1~15 to each class. We decided that we need at least 5000 datasets in order to successfully train the model, thus we handcrafted 400 data for each class, each consisting of data of 20 different soles with the help of Adobe Photoshop. Since, all people cannot mimic the exact given feet position, we gave slight variations (5 variations per class) to the footstep as well as 4 rotations (0, 90, 180, 270 degrees).

Training Set

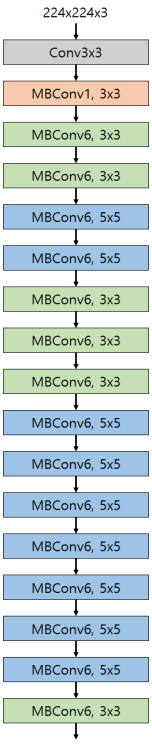


Validation Set



With the help of ImageFolder we were able to transform these images into appropriate tensors and by the built-in functions in DataLoader we split them into training and validation sets. (rate 8:2)

<Model/ Training>



We agreed to make use of the EfficientNet introduced by Mingxing Tan in 2019 for classification, which shows the best performances compared to other models such as ResNet or other recently developed models. (https://arxiv.org/abs/1905.11946) The base model structure has the structure shown on the left. The special part of this model is that it shifts up the model parameters all together (width, heigh, and resolution). We have learned in class and can know by intuition that by increasing the resolution of the input, or scaling the width and depth of the network, we can capture higher features as well as the specifics of the images. However, intuitively we can accept the fact that one of them contributes to one another and is somewhat correlated. Higher resolution might require a deeper and wider network. This model made use of this intuition, where it shifts up the width, height, and resolution of the model proportional to the parameter.

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

We first fix ϕ as 1, assuming twice more resources available, and do a grid search based on the left two equations. Then we try to find that the best corresponding values for the EfficientNet-B0 (the model we use) are α=1.2, β = 1.1, γ = 1.15.

$$\begin{aligned} \max_{d,w,r} \quad & Accuracy\big(\mathcal{N}(d,w,r)\big) \\ s.t. \quad & \mathcal{N}(d,w,r) = \bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\big(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}\big) \\ & \text{Memory}(\mathcal{N}) \leq \text{target\_memory} \\ & \text{FLOPS}(\mathcal{N}) \leq \text{target\_flops} \end{aligned}$$

To get more complex models, we fix the three parameters and then scale up the baseline network with a different ϕ with the equations on the left getting the EfficientNet-B1 ~7, which shows optimal performances.

We decided to use the EfficientNet-B0 model, which has the least parameters (5.3M), due to the lack of training sets (6000) and small number of classes (15). We fine-tuned[2] the pre-trained EfficientNet (with the ImageNet datasets), tuning them with the 6000 footsteps data that we made for the model.

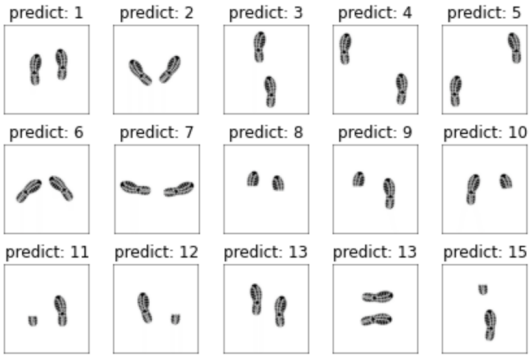
<Results: Model>

```
Epoch 0:
----------
train Loss: 1.12 Acc: 56.9
valid Loss: 4.07 Acc: 35.1
Epoch 1:
----------
train Loss: 0.23 Acc: 92.3
valid Loss: 2.17 Acc: 68.8
Epoch 2:
----------
train Loss: 0.11 Acc: 96.2
valid Loss: 0.58 Acc: 85.7
Epoch 3:
----------
train Loss: 0.06 Acc: 98.0
valid Loss: 0.18 Acc: 94.9
Epoch 4:
----------
train Loss: 0.04 Acc: 98.8
valid Loss: 0.08 Acc: 97.3
Epoch 5:
----------
train Loss: 0.02 Acc: 99.5
valid Loss: 0.08 Acc: 97.2
Epoch 6:
----------
train Loss: 0.01 Acc: 99.7
valid Loss: 0.01 Acc: 99.8
Epoch 7:
----------
train Loss: 0.00 Acc: 99.9
valid Loss: 0.00 Acc: 99.9
Epoch 8:
----------
train Loss: 0.00 Acc: 100.0
valid Loss: 0.00 Acc: 100.0
Epoch 9:
----------
train Loss: 0.00 Acc: 99.9
valid Loss: 0.00 Acc: 100.0
```

The training results were perfect, where it showed 99.9~100% accuracy after a few epochs. We believe that the model was pre-trained to classify high dimensional data, and a black and white 15 class footstep data is relatively very light-loaded for the model.

predictions for the validation sets

We kept in mind that the validation sets are split randomly within the generated datasets, thus the structure of the pictures in the validation set might resemble those in the training set. This might cause a serious issue for the model, since it indicates that the model is prone to overfitting. In order to check this, we tested the model putting newly generated data as inputs. The predictions showed 100% accuracy, therefore being able to state that the model works optimally.



prediction results for newly generated data

The current model assumes that all inputs are correct inputs, in other words, all the inputs will be classified to some class even though they are wrong. It would be best if we could classify all the wrong data precisely and put it into another class, but there were too many variations, which were impossible for us to handle. We decided to solve this problem by classifying the predictions to each class only when the probability (calculated by applying the softmax function to the results) is bigger than 0.8. If the probability is smaller than the baseline, we classify it to class 0 (null class), which we will consider as a wrong input.

[2] **Novel Contribution**: We used the method of transfer learning, using the pre-trained EfficientNet-B0 Model. The model was fine-tuned with 6000 data that we hand-crafted via Adobe Photoshop.

predictions classified to a null class

<Results: Frontend Development>

We created a brief animation, where it shows our system thoroughly.
https://drive.google.com/file/d/1dCg7lTh91f043SztLTTLEo2Bq902vu5V/view?usp=sharing

Initially, we targeted to make a website page that allows the users to manipulate the footsteps freely and put the footstep image as the input, returning the predicted results as the output. However, due to the lack of experience in backend, frontend connection, some fatal errors were unsolvable. Thus, we decided to construct a local server page that has all the functionalities. After the user finishes manipulating the feet positions, the user can download the image to the browser-set directory. Then, by executing the python code he/she can get the photo by its name and get the result.

<Applications>

We strongly believe that this project has contributions not only in the model but also in the applications.[3] It is an unprecedented trial, trying to communicate, send signals, and express our opinions via feet with easily obtainable footprint photos.

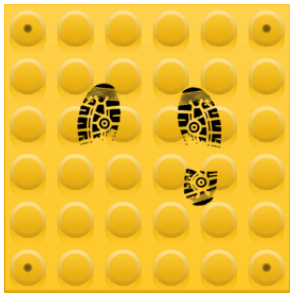
(1) Food and Drink Orders

With the boost of technology, lots of processes are being automated with the help of robots. Some restaurants are making use of robots for serving to reduce cost of labor. We believe that these footstep signal transmission methods will help in simple, food/drink orders in restaurants with predefined categories-to-number matchings. This will help trigger the world of no contact, where people need not call a person or touch the screen to order food. Plus, this will also benefit the customers not in the way of “no contact” but also convenience since they could make simple orders directly without the need to call the waiters/waitresses.

[3] **Novel Contribution:** We believe that these applications are the most essential part, where this **new** idea and **unprecedented** attempt can be applied to various fields giving help to all people on Earth.

(2) Crosswalk Assistance for Blind People

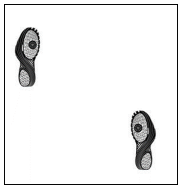
Hard to Find



This system will be of great help to blind people. The government provides a speaking crosswalk for the blind, but only when the people who need help clicks the button. This finding process is very challenging for the blind. We strongly believe that it would be very convenient for them if sensors or tools that can take photos of their feet exist on braille blocks. By assigning some predefined footsteps, which is a sign that a blind person is trying to cross a crosswalk, they could easily express their will to cross it. Blind people are familiar with these blocks more than the existing small button, which would enhance their quality of life.

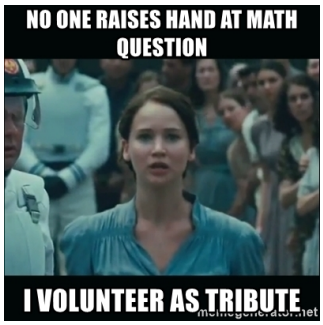
(3) Simple Communication Methods / Signals in Orchestra for Musicians

(I need water)



Moreover, it would hugely benefit the musicians. Musicians, especially those who play wind instruments, usually have their hands full, due to the heavy instruments they are holding. It is quite difficult for them to ask the staff or others for help such as water requests or sheet change. Simple foot language registration with sensors or tools to take photos of feet, will help them to proceed with simple communications.

(4) Assistance in Classes or Discussions in KAIST



Our team tried to find where this “feet language” will benefit students and professors in KAIST. Most Korean students find it difficult to answer simple questions in classes, since they are shy to speak in front of the public. Even for simple questions, there is an awkward silence when the professor asks a question to students. This sometimes frustrates the professor, since he/she thinks that the students aren’t concentrating on his/her lecture. We can prevent these harsh circumstances with the introduction of “Happy Feet”, where students can answer simple questions by their feet. Feet are relatively unshown to others when people are sitting down, which could enlist full participation from students. These characteristics of feet could be used in simple polls in discussions, in which feet can partially secure anonymity.

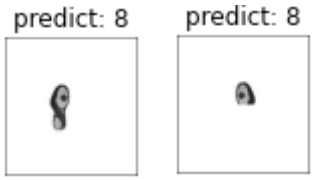
(5) Expansion to Games

One of the most loved games in arcades is the pump game, where people step the corresponding buttons shown on the screen. With the help of “Happy Feet” we believe that we can play these games at home by buying a sensor. Moreover, with the development of VR technology and motion sensors, games are becoming more dynamic and motion-related. We look forward that our model can be expanded to this field, where users can utilize their feet when playing games with the help of “Happy Feet”, which will boost the dynamic and realistic aspect of VR games.

<Discussions>

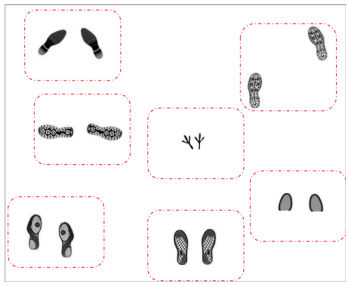
Despite the positives, there are some limitations in our model that we target to improvise in the near future.

(1) Filtering Wrong Shapes



Though some wrong inputs are currently filtered, the algorithm of classifying wrong inputs when probability is smaller than 0.8 isn’t perfect. We can see that some wrong inputs are still predicted properly, since the model mispercepts this as a correct input. We believe that this is happening because of the lack of data. If we introduce this in real life and get access to the big data related to these footsteps. We strongly believe that we can train the model so that it has an extra class, so that the model itself could classify wrong input data by itself, instead of the current method.

(2) Classification of Multiple Footsteps



It would be even better if we the model could classify multiple footsteps and find the expressions for each individual, so that it could be widely used in elevators for multiple users. We found out that this could be done with the YOLO+ResNet or adding features of bounding box regression to the current model, but was not able to do this in the project due to the heavy load of data generation. We also believe that this could be done if data could be collected on a large scale more easily, without the need for them to be handcrafted.

<Role Distribution>

Jio Oh: Model Training (backend/train.ipynb), Model Research (backend/model.py)
Powerpoint (team16_ppt.pdf) & Report (team16_report.pdf), Presentation

Donghyeon Lee: Model Testing (backend/test.py), Model Research (backend/model.py)
Data Generation (dataset), Figure Creation (footstep results, model)

Jihyeong Hong: Frontend Development (frontend/)
Data Generation(dataset) Animation Creation (frontend animation)