Kvasir Dataset Deep Learning

202034308 김민규
202035103 강예진
202035240 조준우
202035143 남기주

01

Kvasir Dataset

02

Using Models

03

Increase Accuracy

04

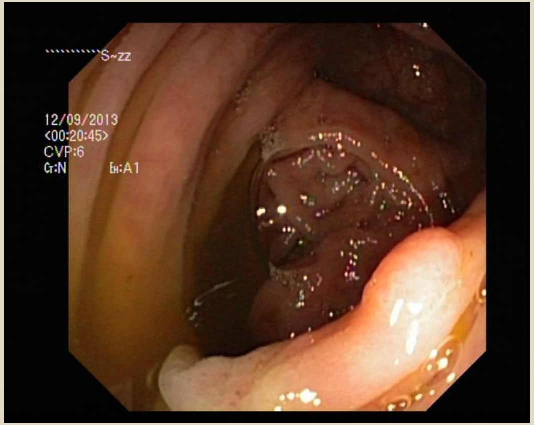
Service

01

Kvasir Dataset

Kvasir Dataset

Dataset dividing the types of diseases identified by endoscopy



Ex) Polyps

Class

1. dyed-lifted-polyps
2. dyed-resection-margins
3. esophagitis
4. normal-cecum
5. normal-pylorus
6. normal-z-line
7. polyps
8. ulcerative-colitis

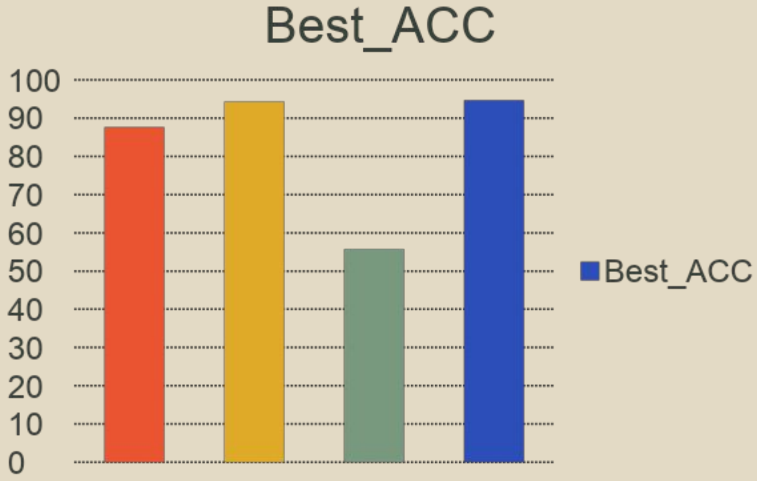
02

Using Models

Using Models

VGGNet	- The structure is much simpler than the existing model - The number of layers goes deeper to 16
ResNet	- Add shortcuts to increase performance as the number of layers increases
ViT	- Processing images without significant changes to Transformer's entire architecture - The disadvantage of having to train a larger amount of datasets than CNN's
Efficientnet	- With AutoML, find the best combination of network depth, channel width, and input image and maximize efficiency with limited resources

Learning Result



VGGNet
87.65 %

ResNet
94.37%

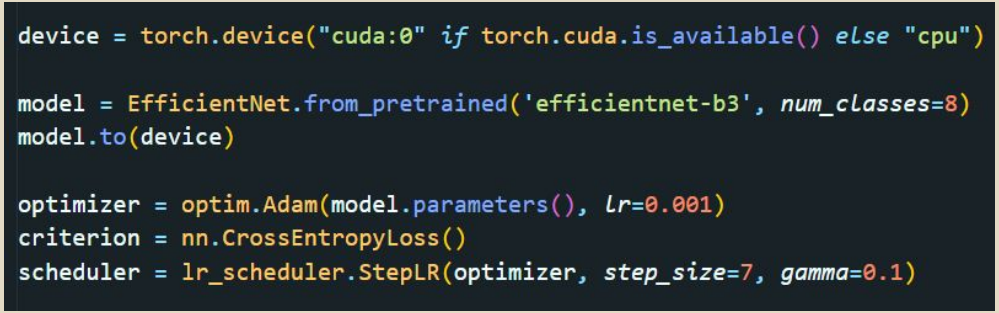
ViT
55.75 %

Efficientnet
94.69%

Batch_Size = 16, Train : Val = 8 : 2
Epoch = 30, Total Dataset = 8,000

Select Model

“ Efficientnet – b3”



```
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

model = EfficientNet.from_pretrained('efficientnet-b3', num_classes=8)
model.to(device)

optimizer = optim.Adam(model.parameters(), lr=0.001)
criterion = nn.CrossEntropyLoss()
scheduler = lr_scheduler.StepLR(optimizer, step_size=7, gamma=0.1)
```

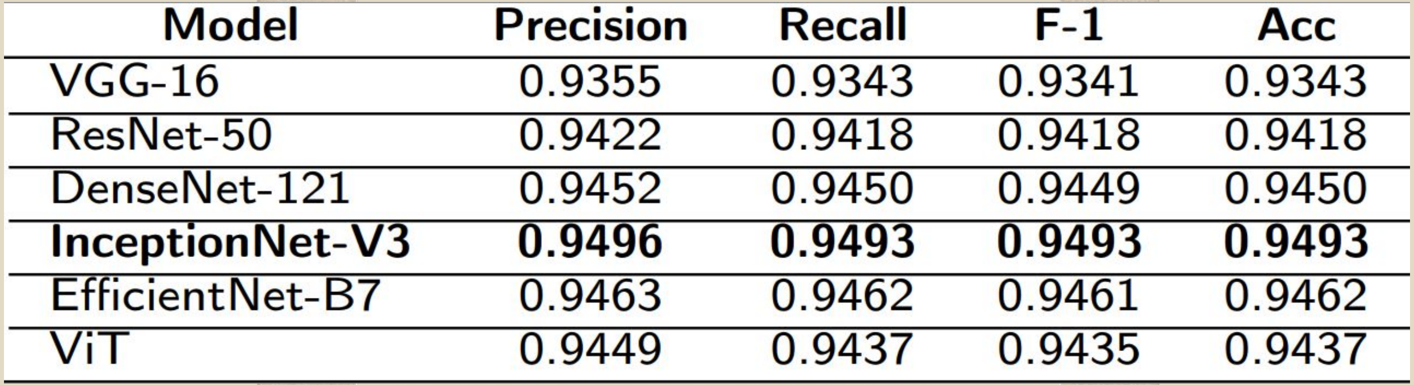
Only use code that automatically finds the optimal value of learning_rate

03

Increase Accuracy

Goal

Beyond current best performance models



Model	Precision	Recall	F-1	Acc
VGG-16	0.9355	0.9343	0.9341	0.9343
ResNet-50	0.9422	0.9418	0.9418	0.9418
DenseNet-121	0.9452	0.9450	0.9449	0.9450
InceptionNet-V3	**0.9496**	**0.9493**	**0.9493**	**0.9493**
EfficientNet-B7	0.9463	0.9462	0.9461	0.9462
ViT	0.9449	0.9437	0.9435	0.9437

Highest performance model today

Methods

DCGANS

Using Deep Convolution Gan to
Increasing the dataset using

Augmentation

Improve image recognition accuracy through image aggregation

Data Processing

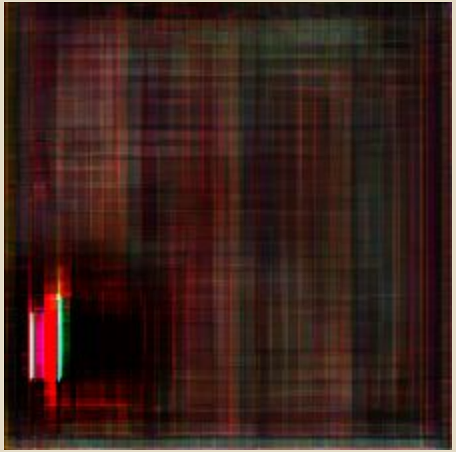
Pre-processing to the best data format for the model to handle

Deep Convolutional GAN

생성자, 판별자의 신경망을 가지고 진짜 같은 가짜를 만들어 냄

Feature	Used code	Description
Max Pooling To Strided Convolution	nn.conv2D()	Replace image pixels with a combination of surrounding pixels
Eliminate Fully-Connected Layers	Eliminate Fully-Connected Layers in Generator	Except for the last softmax layer to judge the output result, all FC layers are excluded
Batch-Normalization	BatchNormalization()	The process of being included within a neural network to adjust the mean and variance when learning

Result



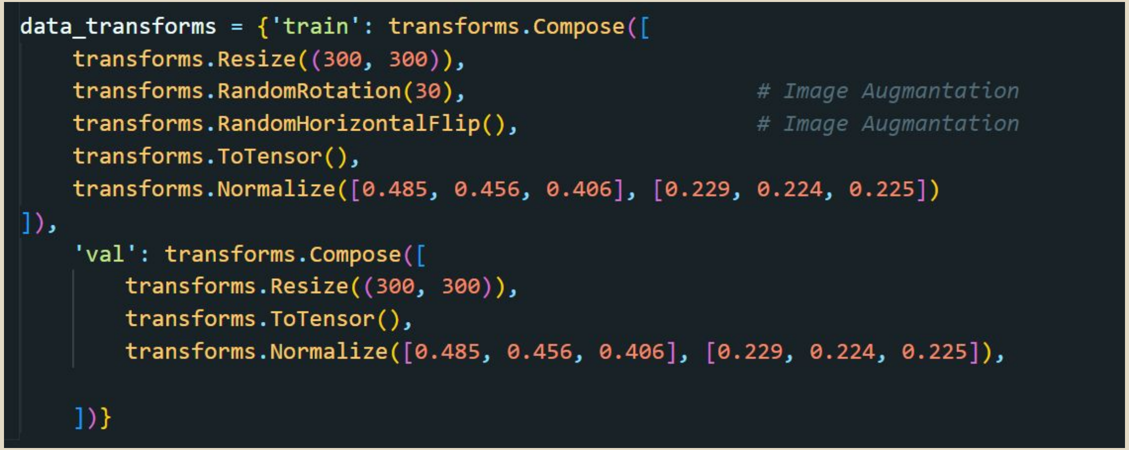
DCGAN의 결과물 중 하나

Photo noise is strong and the difference from existing images is too high to be used

Augmantation

- Flip the image up, down, left and right to free up new training data, Noise by cutting, etc. to secure new image data
- Increase image recognition accuracy and improve overfitting problems

Code (with Data Processing)



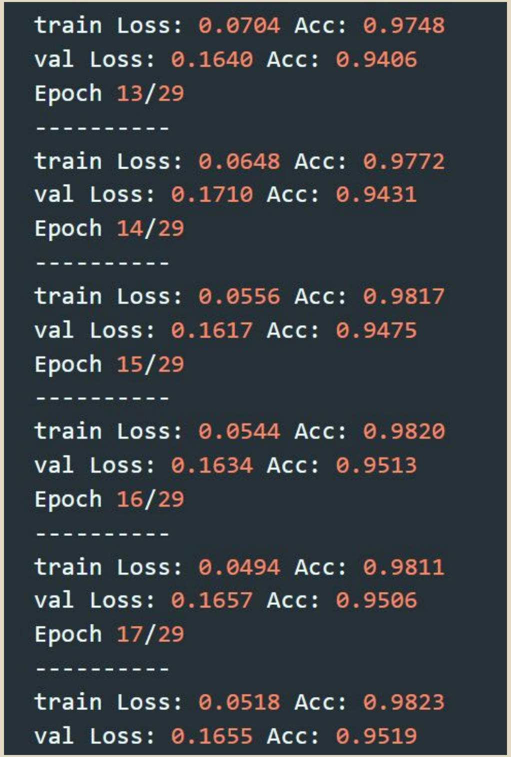
```
data_transforms = {'train': transforms.Compose([
    transforms.Resize((300, 300)),
    transforms.RandomRotation(30),                        # Image Augmantation
    transforms.RandomHorizontalFlip(),                    # Image Augmantation
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
]),
    'val': transforms.Compose([
        transforms.Resize((300, 300)),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]),

    ])}
```

- **Transforms.RandomRotation()**
  - ✔ **Rotates randomly according to the given angle.**

- **Transforms. RandomHorizontalFLip ()**
  - ✔ **Turn it horizontally at random.**

- **Resize and preprocess with the corresponding image because the recommended input size of the model is 300 * 300**

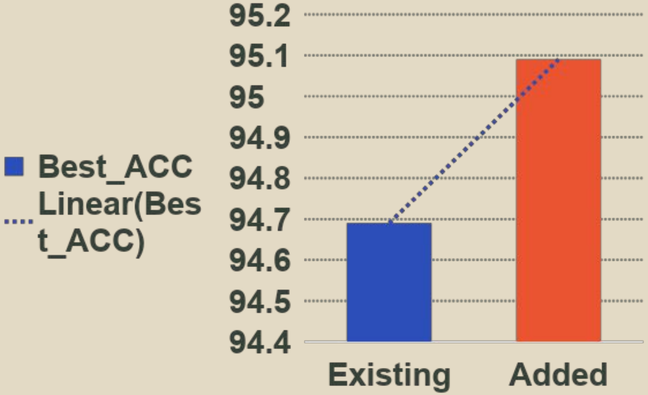
Result



```
train Loss: 0.0704 Acc: 0.9748
val Loss: 0.1640 Acc: 0.9406
Epoch 13/29
----------
train Loss: 0.0648 Acc: 0.9772
val Loss: 0.1710 Acc: 0.9431
Epoch 14/29
----------
train Loss: 0.0556 Acc: 0.9817
val Loss: 0.1617 Acc: 0.9475
Epoch 15/29
----------
train Loss: 0.0544 Acc: 0.9820
val Loss: 0.1634 Acc: 0.9513
Epoch 16/29
----------
train Loss: 0.0494 Acc: 0.9811
val Loss: 0.1657 Acc: 0.9506
Epoch 17/29
----------
train Loss: 0.0518 Acc: 0.9823
val Loss: 0.1655 Acc: 0.9519
```

Increase accuracy to 95.19% to exceed peak model performance

Compare Analysis



Efficientnet
94.69%

↓

Image Augmentation & **Data Processing**

↓

Efficientnet
95.19%

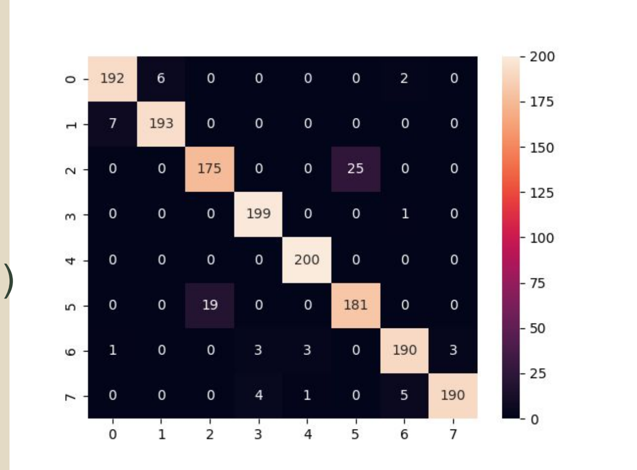
Verification

A new model made by the team(**Efficientnet-b3**)

	precision	recall	f1-score
score	0.9507	0.9506	0.9505

Existing Best Performance Models(**Inceptionnet-V3**)

	precision	recall	f1-score
score	0.9496	0.9493	0.9493



	0	1	2	3	4	5	6	7
0	192	6	0	0	0	0	2	0
1	7	193	0	0	0	0	0	0
2	0	0	175	0	0	25	0	0
3	0	0	0	199	0	0	1	0
4	0	0	0	0	200	0	0	0
5	0	0	19	0	0	181	0	0
6	1	0	0	3	3	0	190	3
7	0	0	0	4	1	0	5	190

Confusion-Matrix

Method : The average of 10 best_weight weight files
after 10 training with the same code

04

Service

Process

Model Load

Service with the weighted file of the previously learned Efficientnet model

Analysis

Deep learning models use available web frameworks to learn and predict new images

Show Result

Derive predicted images and classes (8 classes previously introduced)

Service Purpose & User

Purpose

- Health insurance workers find it difficult to determine whether the data that came in for insurance claims is false or not

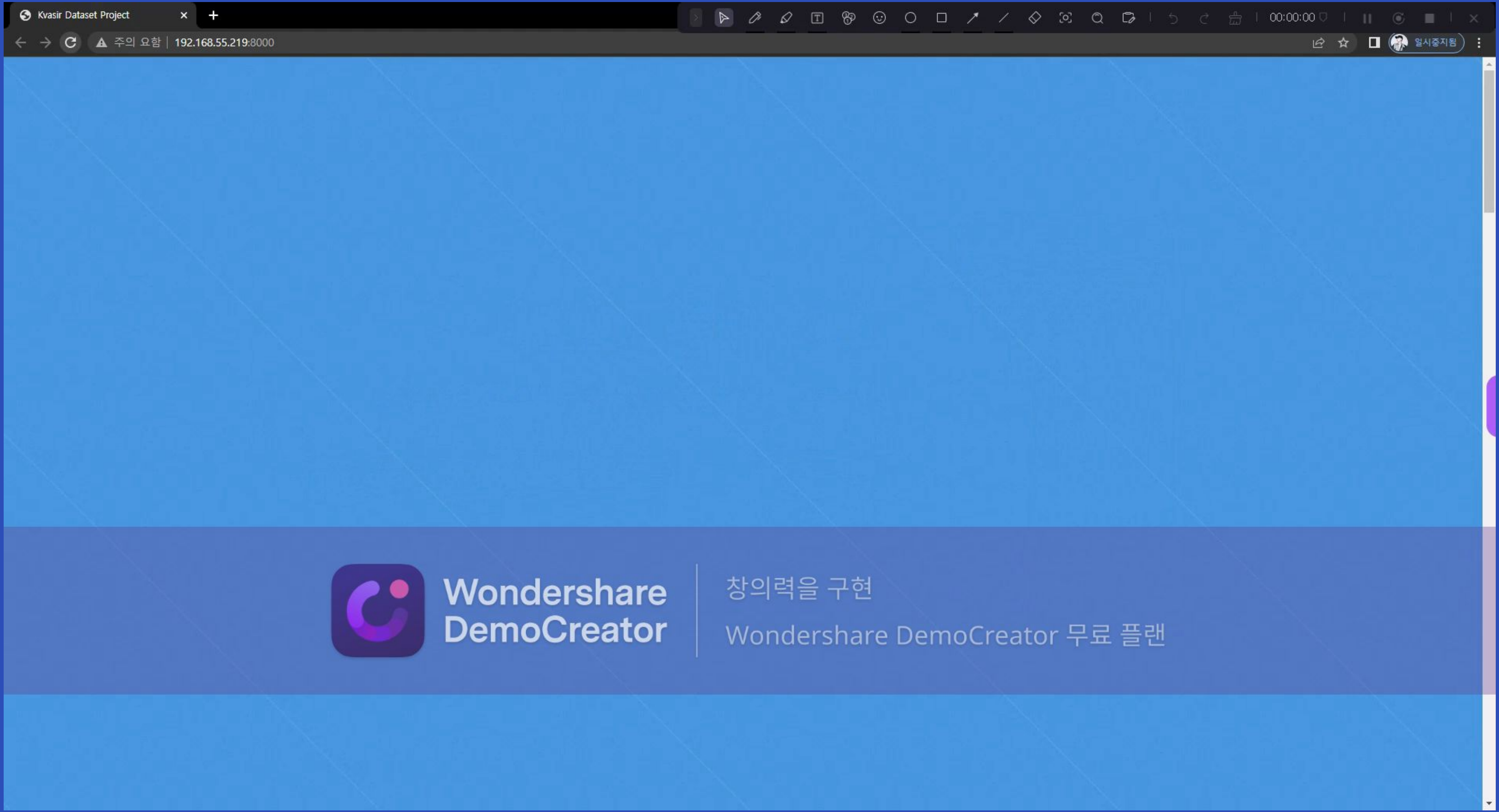
- This service can help you determine what endoscopic data is sent by claimants or whether it is false or not

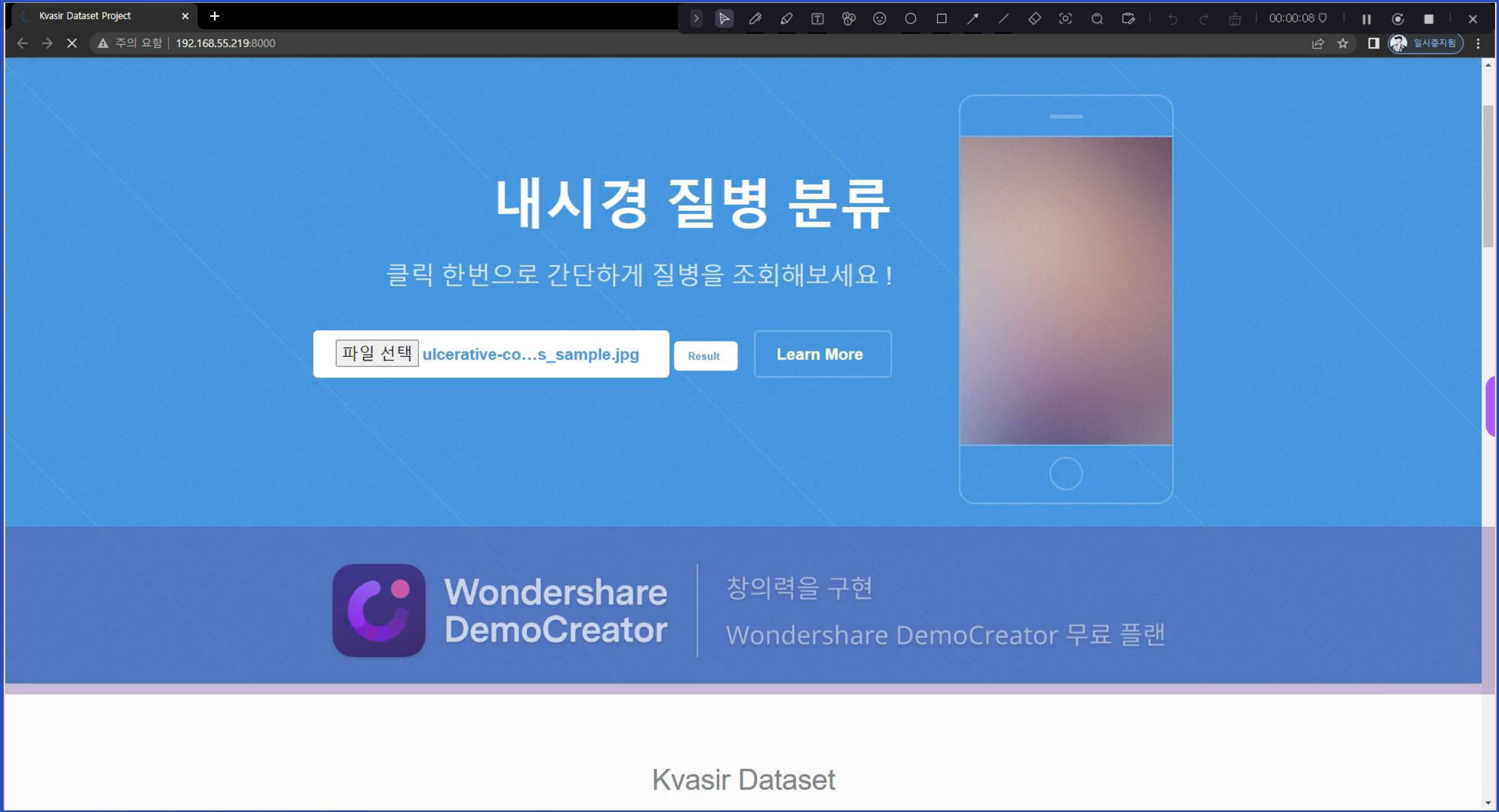
User

- insurance related workers

Tools

- The default language is Python
- Using deep learning models trained with the Flask web framework for serving and prediction
- Other frameworks for web services through CSS, HTML, and JS


Kvasir Dataset Project
주의 요함 | 192.168.55.219:8000
00:00:00
일시중지됨



내시경 질병 분류

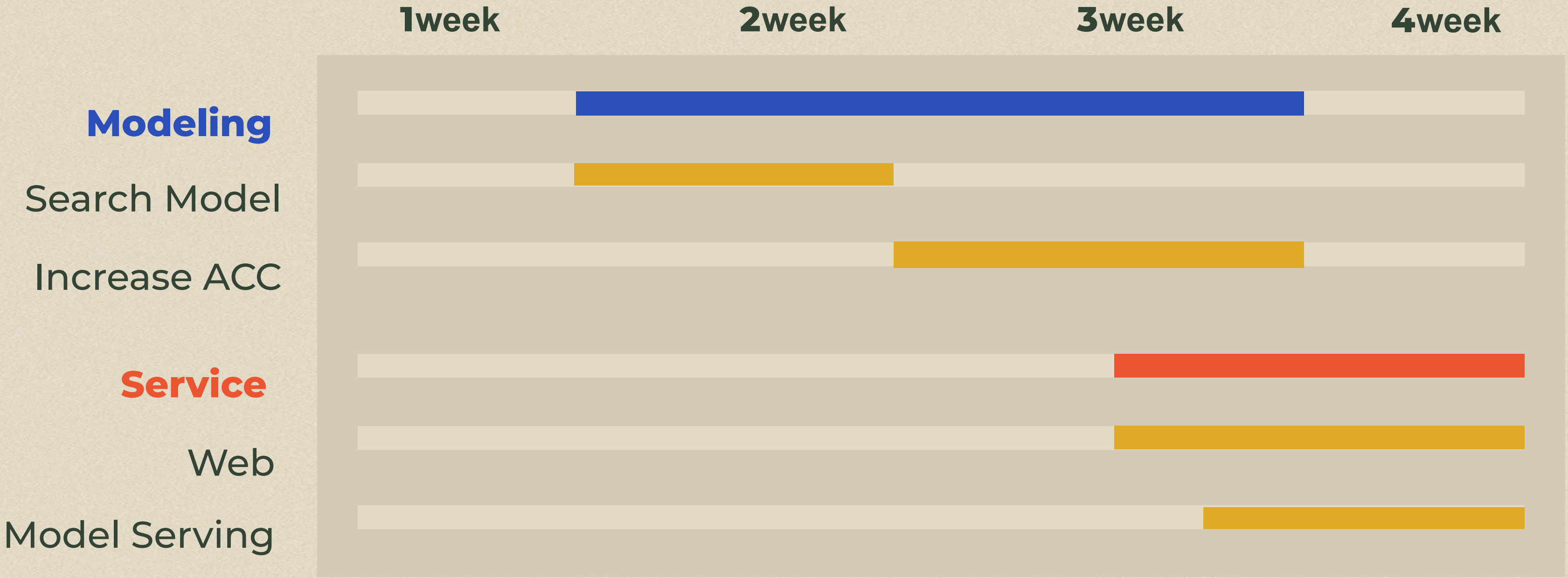
클릭 한번으로 간단하게 질병을 조회해보세요 !

Kvasir Dataset

Future goals

Aim for more sophisticated serviceization

Schedule


1week
2week
3week
4week
Modeling
Search Model
Increase ACC
Service
Web
Model Serving

THANKS