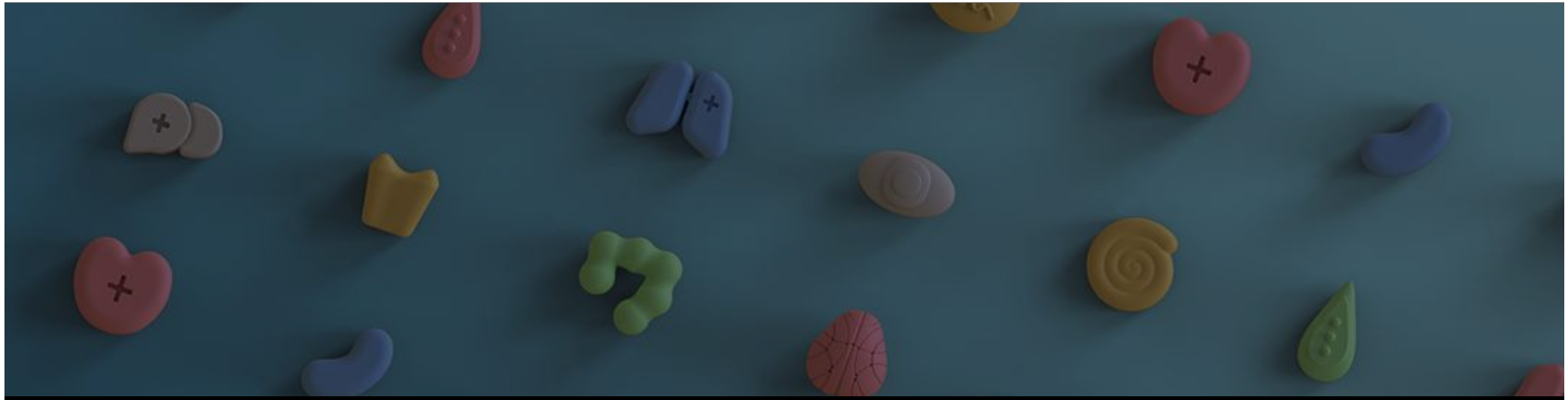
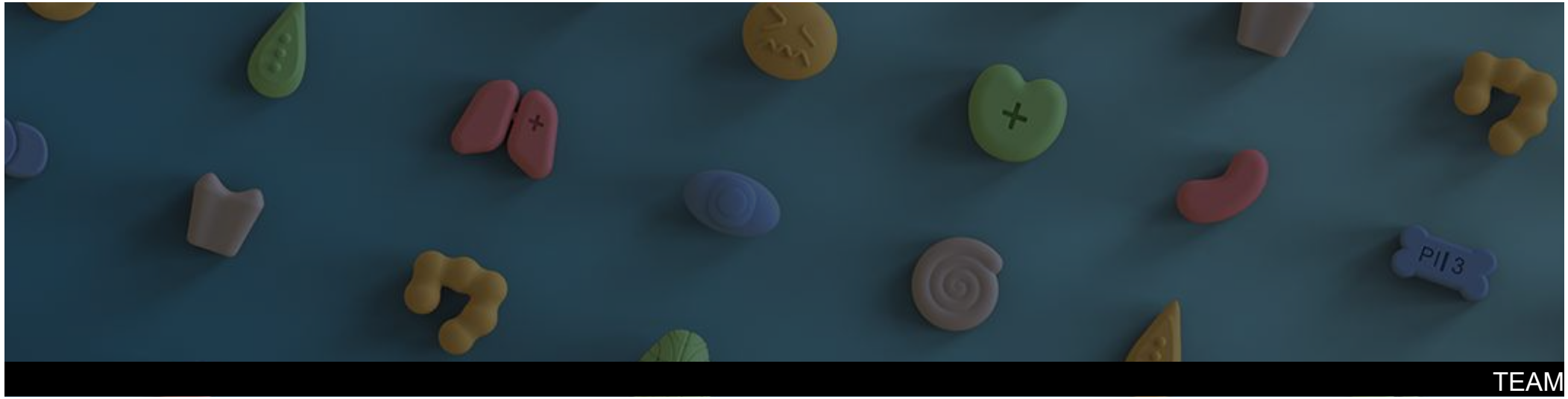


PILL RECOGNITION MODEL USING DEEP LEARNING



TEAM

TEAM

허혁

황나현

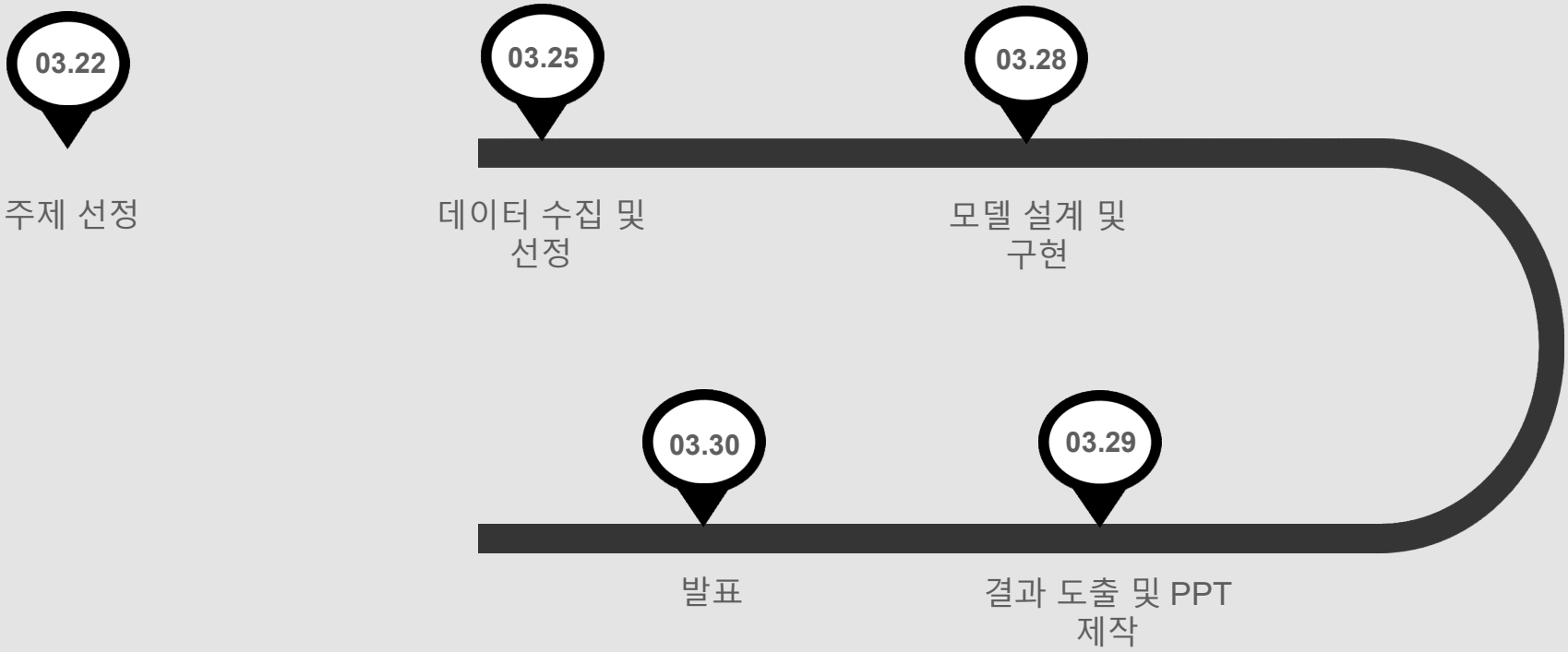
하상준

허정민

현지연

CONTENTS

PROJECT SCHEDULE



CONTENTS

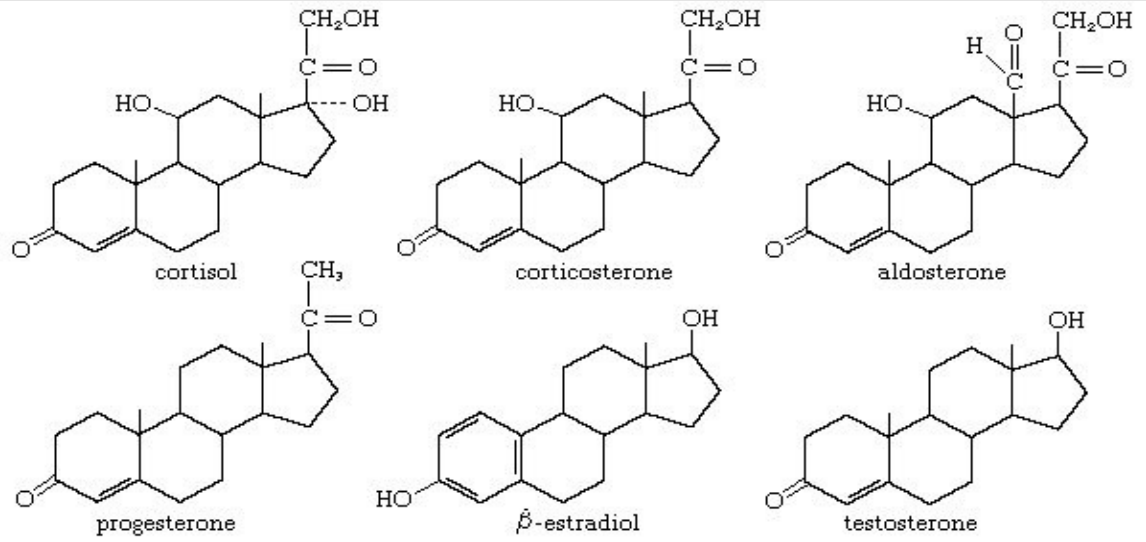
TOPIC

What is PILL RECONGNITION MODEL USING DEEP LEARNING?

DEEP LEARNING CNN기술을 활용하여 스테로이드 알약과 비스테로이드 알약 분류하기

TOPIC

STEROID



TOPIC

STEROID causes of side effects

1. 정보부제

일반의약품 표면에 '스테로이드'라고 명시되어 있지 않다.

2. 잘못된 분류

국내 스테로이드제 상당수가 일반의약품으로 분류돼 있다.

3. 낮은 인식도

스테로이드에 대한 일반인들의 인식도가 낮은 편이다.

4. 과다처방

빠른 효과를 보기 위해 의사들이 스트로이드제를 과다 처방한다.

TOPIC

STEROID causes of side effects

야간용 1정(610.9mg)중

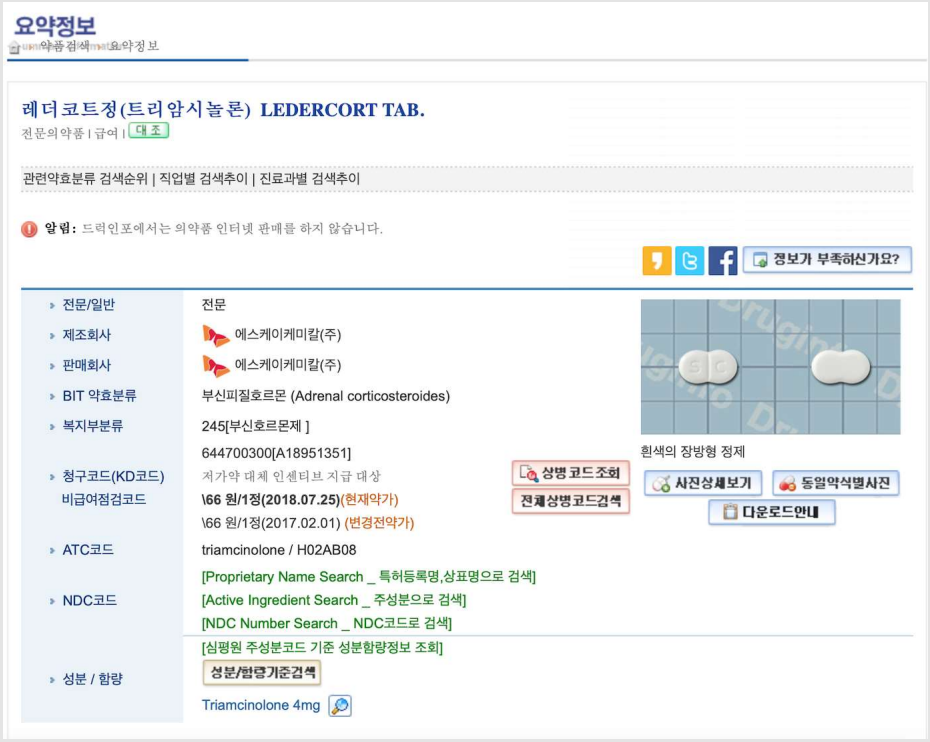
Acetaminophen 200mg

Dl-Chlorpheniramine maleate 1.25mg

TOPIC

STEROID misuse cases

DATA Preprocessing



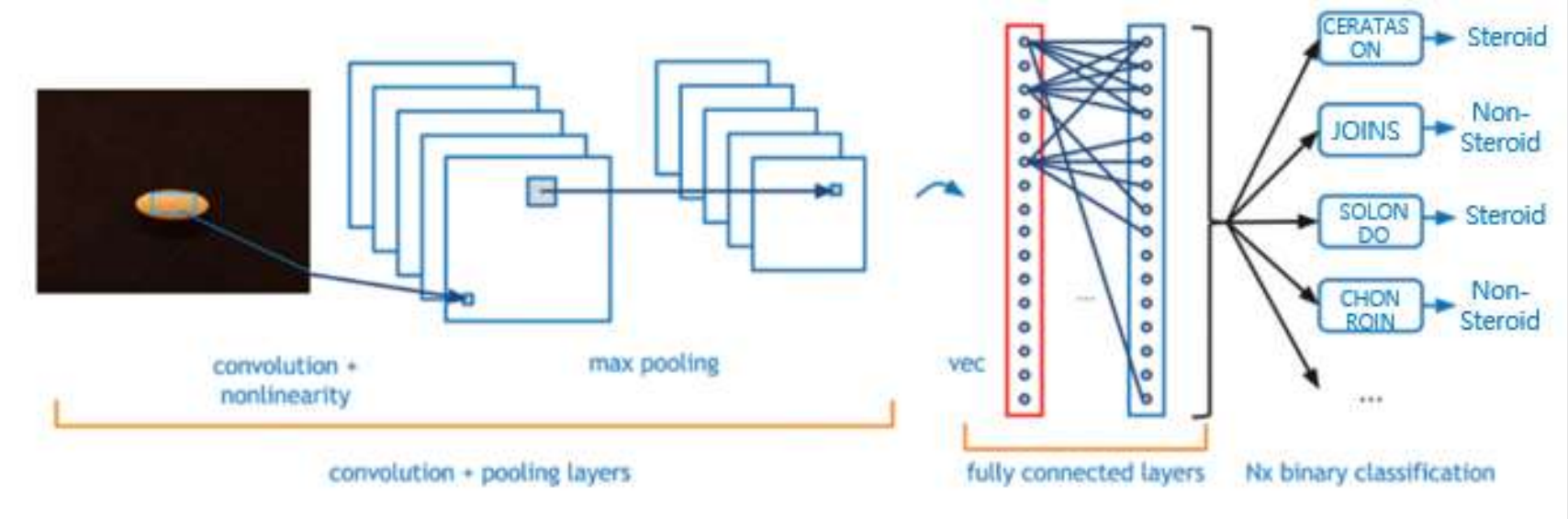
https://www.druginfo.co.kr

수집된 알약 데이터

분 류	성분	제조&판매회사
Steroid	JR HYDROCORTISONE TAB 5mg 제이알히드로코르티손정 5mg	(주) 비보존 제약
	LEDERCORT TAB 레더코트정	에스케이케미칼(주)
	TRACINONE TABS 트라시논정	초당약품공업(주)
	CERETASONE TAB 트라시논정	건일제약(주)
	SOLONDO 소론도정(프레드니솔론)	(주)유한양행
Non_Steroids	SAMJIN CHONROIN 콜로인캡슐	삼진제약(주)
	JOINS TAB 조인스정	에스케미칼(주)

DATA Preprocessing

CNN (Convolution Neural Networks)



"이미지의 공간 정보를 유지한 상태로 학습이 가능한 모델"

CONTENTS

MODELING PROCESS

Input Image(128x128x3)

VGG16(4x4x512)

Convolution Layer(8192)
Relu Layer
Dropout

Convolution Layer
Relu Layer
Dropout

Convolution Layer
Softmax

Output (10 Class)

What is this?

Non-Steroid

MODELING

MODELING

```
vgg16 = VGG16(weights = 'imagenet', include_top =
       False, input_shape = (128, 128, 3))
```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 128, 128, 3)]	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147584
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0

```
Total params: 14,714,688
Trainable params: 7,079,424
Non-trainable params: 7,635,264
```

MODELING

```
model.compile(loss='categorical_crossentropy',
                    optimizer="adam",
                   metrics=['accuracy'])
```

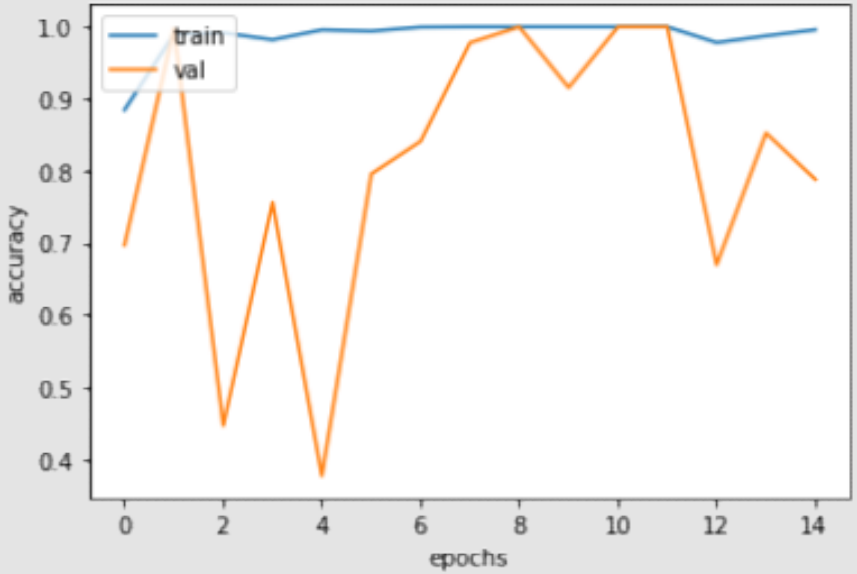
Model: "sequential"

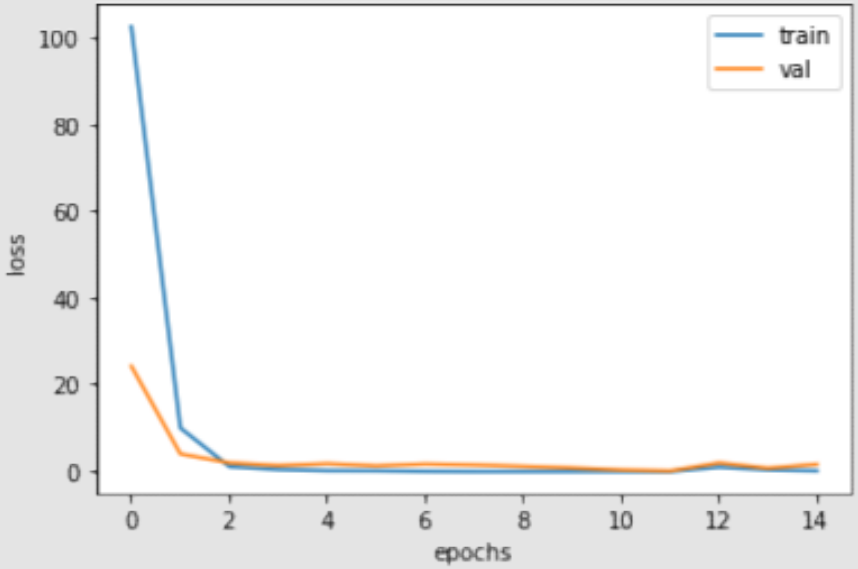
Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 4, 4, 512)	14714688
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 256)	2097408
batch_normalization (BatchNo	(None, 256)	1024
activation (Activation)	(None, 256)	0
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 256)	65792
batch_normalization_1 (Batch	(None, 256)	1024
activation_1 (Activation)	(None, 256)	0
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 10)	2570
activation_2 (Activation)	(None, 10)	0

```
Total params: 16,882,506
Trainable params: 9,246,218
Non-trainable params: 7,636,288
```

```
47/47 [==============================] - 10s 212ms/step - loss: 3.0063 - accuracy: 0.9970 - val_loss: 2.4301 - val_accuracy: 0.7631
Epoch 4/20
47/47 [==============================] - 10s 216ms/step - loss: 0.3849 - accuracy: 0.9948 - val_loss: 2.3625 - val_accuracy: 0.6313
Epoch 5/20
47/47 [==============================] - 10s 212ms/step - loss: 0.6515 - accuracy: 0.9778 - val_loss: 1.7096 - val_accuracy: 1.0000
Epoch 6/20
47/47 [==============================] - 10s 209ms/step - loss: 0.0784 - accuracy: 1.0000 - val_loss: 1.6062 - val_accuracy: 0.9111
Epoch 7/20
47/47 [==============================] - 10s 206ms/step - loss: 0.0358 - accuracy: 1.0000 - val_loss: 1.7061 - val_accuracy: 0.9960
Epoch 8/20
47/47 [==============================] - 10s 204ms/step - loss: 0.0323 - accuracy: 1.0000 - val_loss: 1.4451 - val_accuracy: 0.9995
Epoch 9/20
47/47 [==============================] - 10s 204ms/step - loss: 0.0606 - accuracy: 0.9980 - val_loss: 1.8623 - val_accuracy: 0.8157
Epoch 10/20
47/47 [==============================] - 10s 206ms/step - loss: 0.8324 - accuracy: 0.9630 - val_loss: 1.2183 - val_accuracy: 0.5005
Epoch 11/20
47/47 [==============================] - 10s 208ms/step - loss: 0.8356 - accuracy: 0.9724 - val_loss: 0.7939 - val_accuracy: 0.8803
Epoch 12/20
47/47 [==============================] - 10s 208ms/step - loss: 0.4017 - accuracy: 0.9970 - val_loss: 4.3230 - val_accuracy: 0.3975
Epoch 13/20
47/47 [==============================] - 10s 208ms/step - loss: 1.1523 - accuracy: 0.9869 - val_loss: 1.0357 - val_accuracy: 0.9929
Epoch 14/20
47/47 [==============================] - 10s 208ms/step - loss: 0.2260 - accuracy: 0.9983 - val_loss: 1.5904 - val_accuracy: 0.5540
Epoch 15/20
47/47 [==============================] - 10s 208ms/step - loss: 0.0714 - accuracy: 0.9997 - val_loss: 1.4840 - val_accuracy: 0.4566
Epoch 16/20
47/47 [==============================] - 10s 206ms/step - loss: 0.0278 - accuracy: 1.0000 - val_loss: 1.7093 - val_accuracy: 0.4030
Epoch 17/20
47/47 [==============================] - 10s 207ms/step - loss: 0.0166 - accuracy: 1.0000 - val_loss: 1.6253 - val_accuracy: 0.4404
Epoch 18/20
47/47 [==============================] - 10s 207ms/step - loss: 0.0404 - accuracy: 1.0000 - val_loss: 0.9581 - val_accuracy: 0.8505
Epoch 19/20
47/47 [==============================] - 10s 207ms/step - loss: 0.0097 - accuracy: 1.0000 - val_loss: 0.7464 - val_accuracy: 0.9020
```

MODELING VISUALIZATION





MODELING ERROR

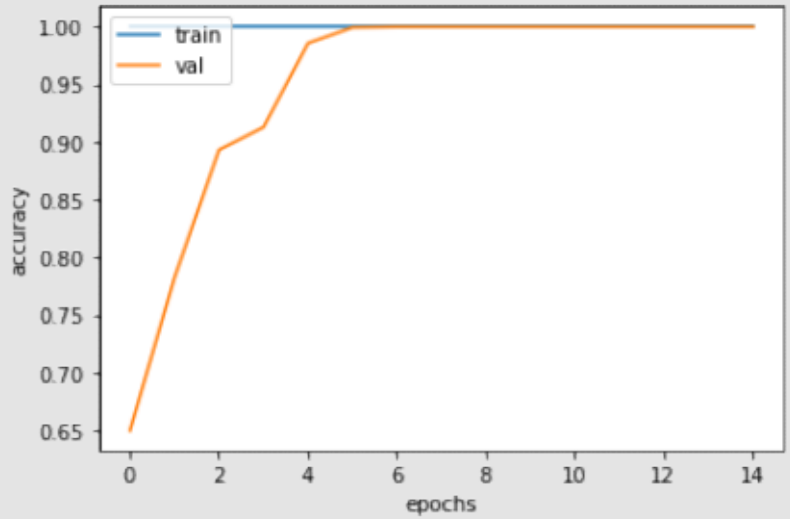
MODELING IMPROVEMENT

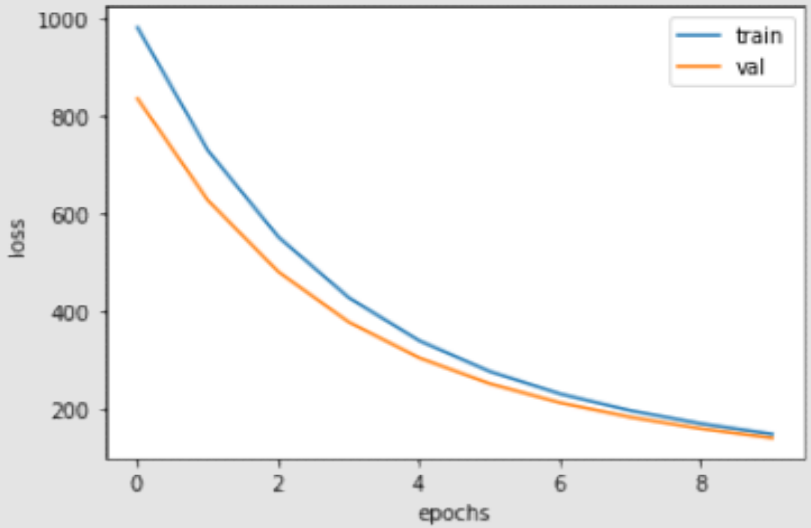
1. 이상치 데이터 제거
2. 에폭 순차적 조정 (15-> 10 -> 7)
3. EarlyStopping
4. L2 정규화 순차적 조정
5. Dropout 순차적 조정

```
from tensorflow.python.keras.callbacks import EarlyStopping

callbacks = EarlyStopping(monitor = 'val_loss', min_delta=0, patience=3, verbose=0)

epochs = 10
batch_size=150

hist = model.fit(x_train, y_train,
                 validation_data=(x_val, y_val),
                 epochs=epochs,
                 batch_size=batch_size,
                 callbacks = [callbacks])
```

CONCLUSION

RESULTS





TEST DATA

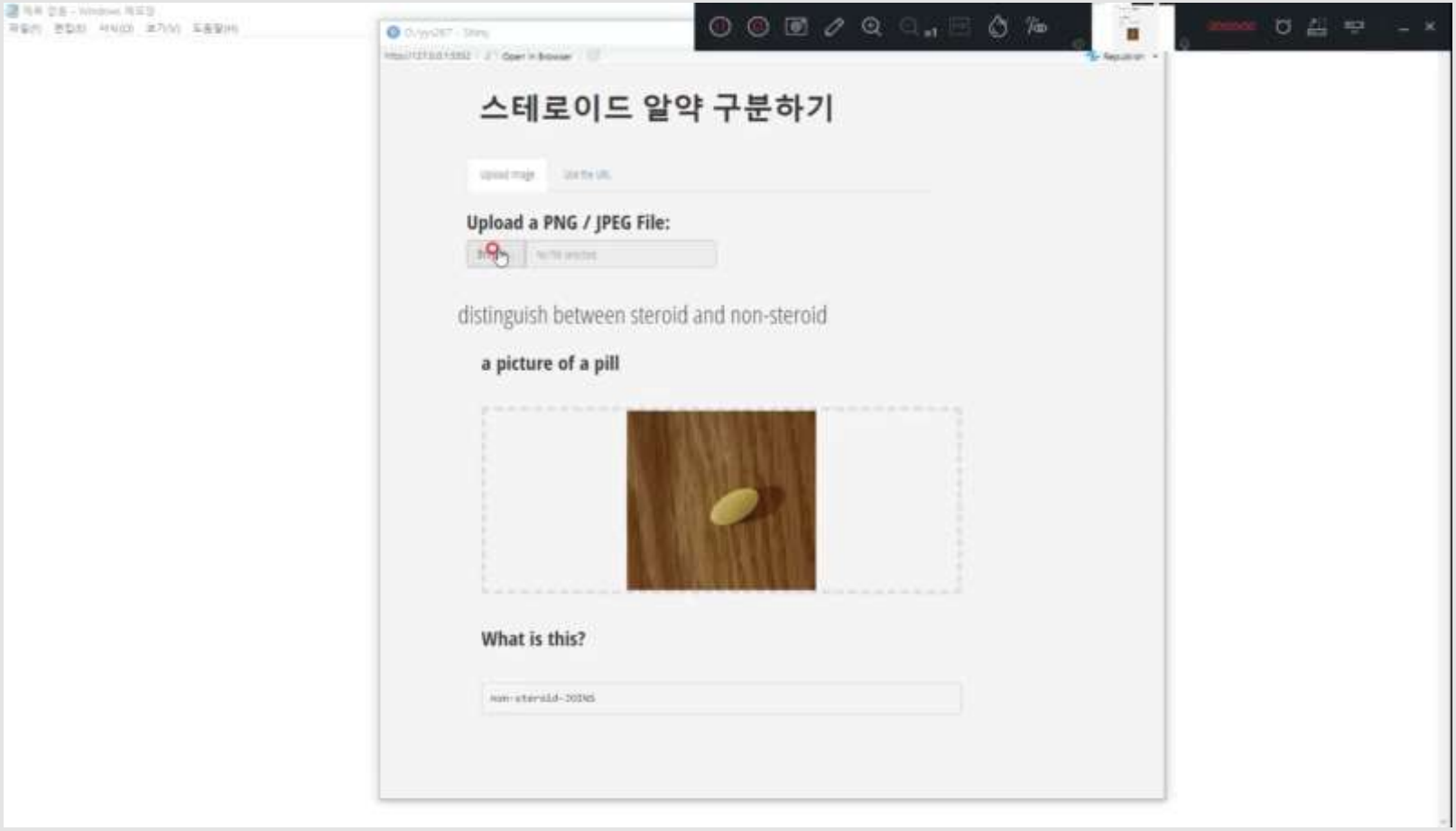
소론도정(solondo-스테로이드)

조인스정(joins-비스테로이드)

콘로인(chonroin-비스테로이드)

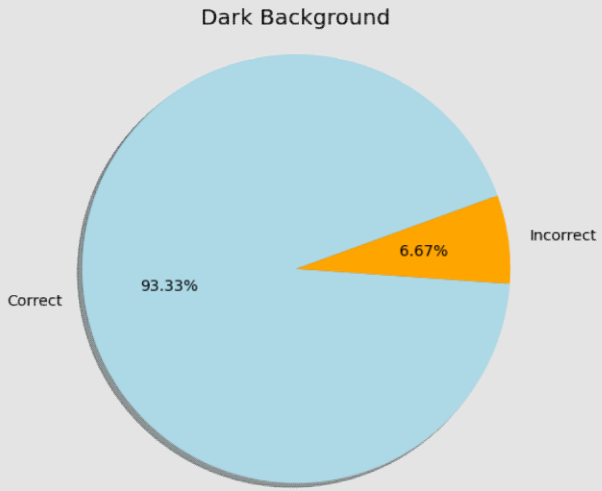
SIMULATION

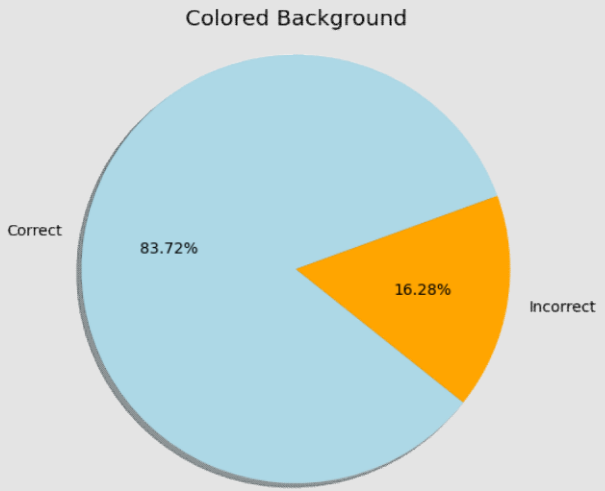
R SHINY



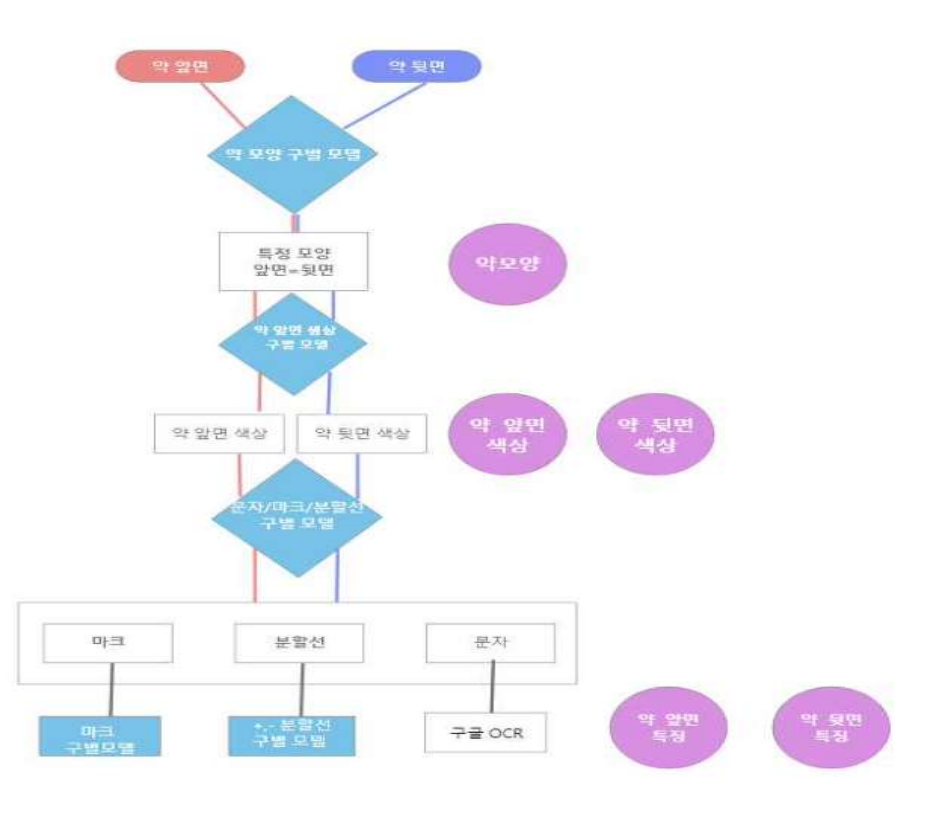
SIMULATION

R SHINY





IMPROVEMENT



1. 추가 데이터 수집으로 알약의 범위 및 범용성 확보

2. OCR 문자인식 및 OpenCV 패키지 내의 함수를 활용한 알약 문자 학습

3. VGG를 기반으로 하는 ResNet 구현

CONCLUSION

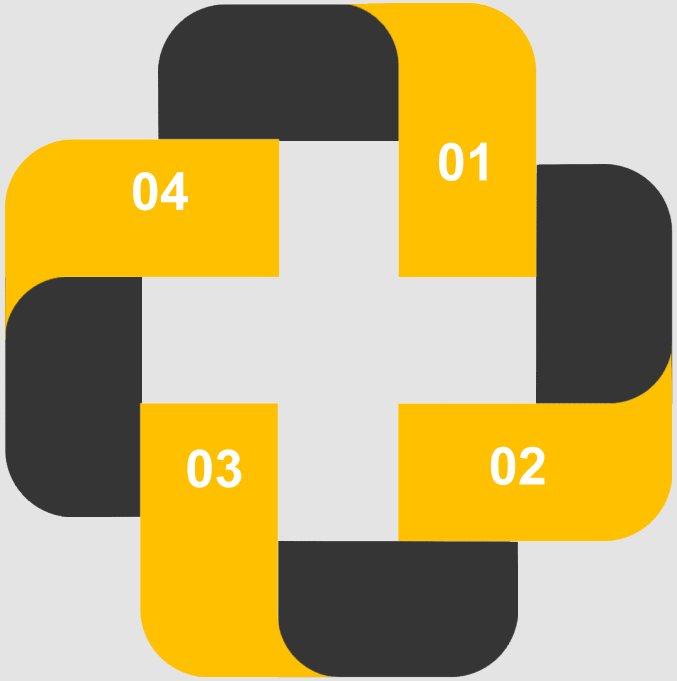
APPLICATION PLAN

01 알약 데이터를 확대하여 모든 알약을 분류해주는 모델로 확대

02 노년층을 타겟팅한 알약 분류 스마트폰 어플리케이션 비스 개발

03 약물 제한이 많은 운동선수를 대상으로 금지약물 분류 모델 서비스 개발

04 임산부를 타겟팅한 유해성분 알약 분류 모델 개발




THANK YOU
TEAM H