

---

박학다식

# Weekly Presentation .3

---

2019311036 신새별

2018311095 장민근

2017313764 김재연

2017314786 정동진

2015313546 김창현

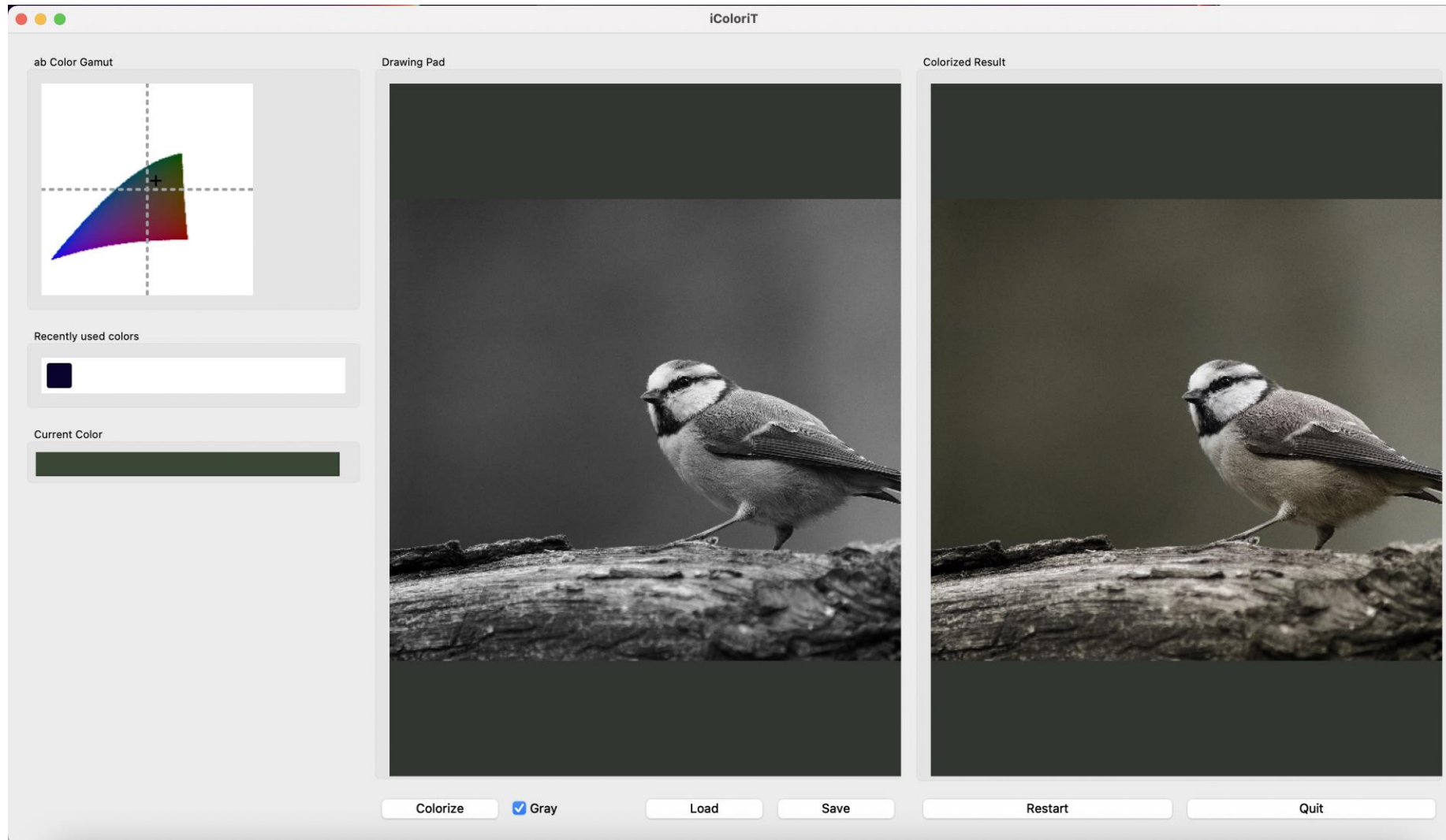
---

# Front-End

---

2015313546 김창현

# 1. Front-End



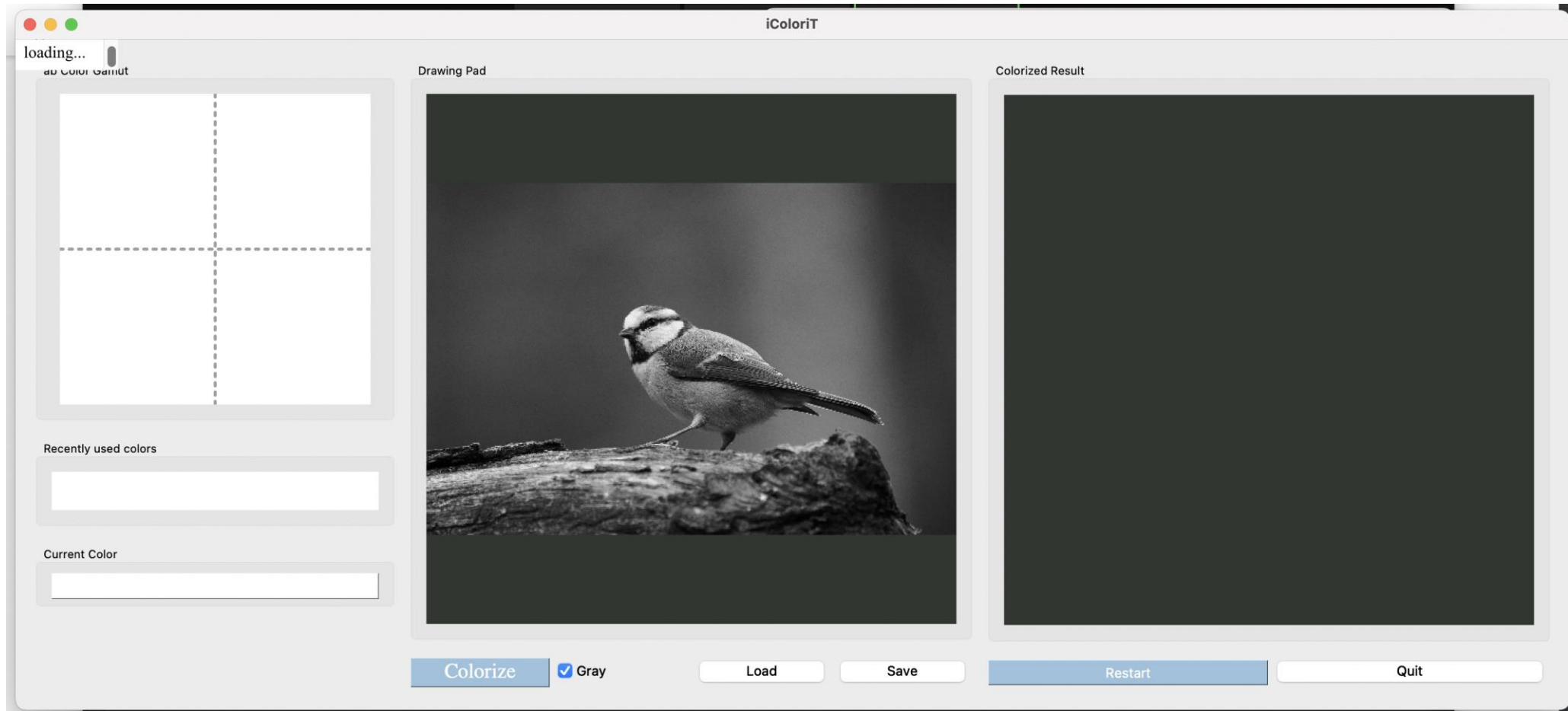
# 1. Front-End

---

- 버튼 크기 및 색상 변경
- 이미지 영역 디자인 변경
- 색 선택 부분 정 가운데로 배치
- 글꼴 수정 및 크기 변경
- App-bar 추가
- Flask

# 1. Front-End

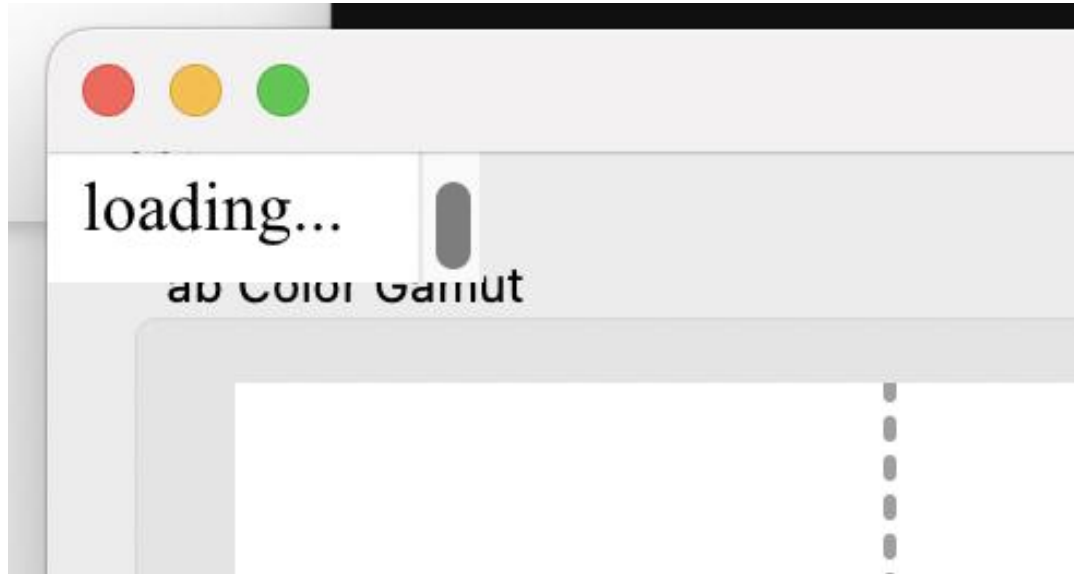
## Flask error



# 1. Front-End

---

## Flask error



Web에 gui가 렌더링되지 않음

Gui 실행창 안에 html파일로 작성했던  
글이 나옴

# 1. Front-End

---

## Next Week

Flask 문제 해결 및 대안 적용( webkit )

모델 쪽 인원 보충 및 프론트엔드 변경 사항 수정

---

# Model

---

2017313764 김재연



# Project Planning

[illegible]

## 2. Model

---

### Noise & Augmentation

#### Noise

224 x 224 Resize or 384 \* 384 Resize  
Gaussian Noise

#### Augmentation

224 x 224 Resize or 384 \* 384 Resize  
밝기, 좌우 반전, 상하 반전

\* 로컬 환경에서 47,807장으로 테스트

## 2. Model

### Noise & Augmentation



Original Image



Var : 0.5



Var : 0.8



Var : 1.0



Var : 1.5

Noise Result

## 2. Model

### Noise & Augmentation



Original Image



Brightness



Horizontal Flip



Vertical Flip

Augmentation Result

## 2. Model

### Noise & Augmentation

#### Time Taken

	224 * 224	384 * 384
Noise	250	535
Augment - 밝기	150	161
Augment - 좌우 반전	80	83
Augment - 상하 반전	128	155

단위 : 초

가장 오래 걸리는 작업(535초)을 기준으로 하였을 때,  
157만장 가량의 프로세싱은 5시간 내로 끝날 것으로 예상

## 2. Model

---

### Trial and Error

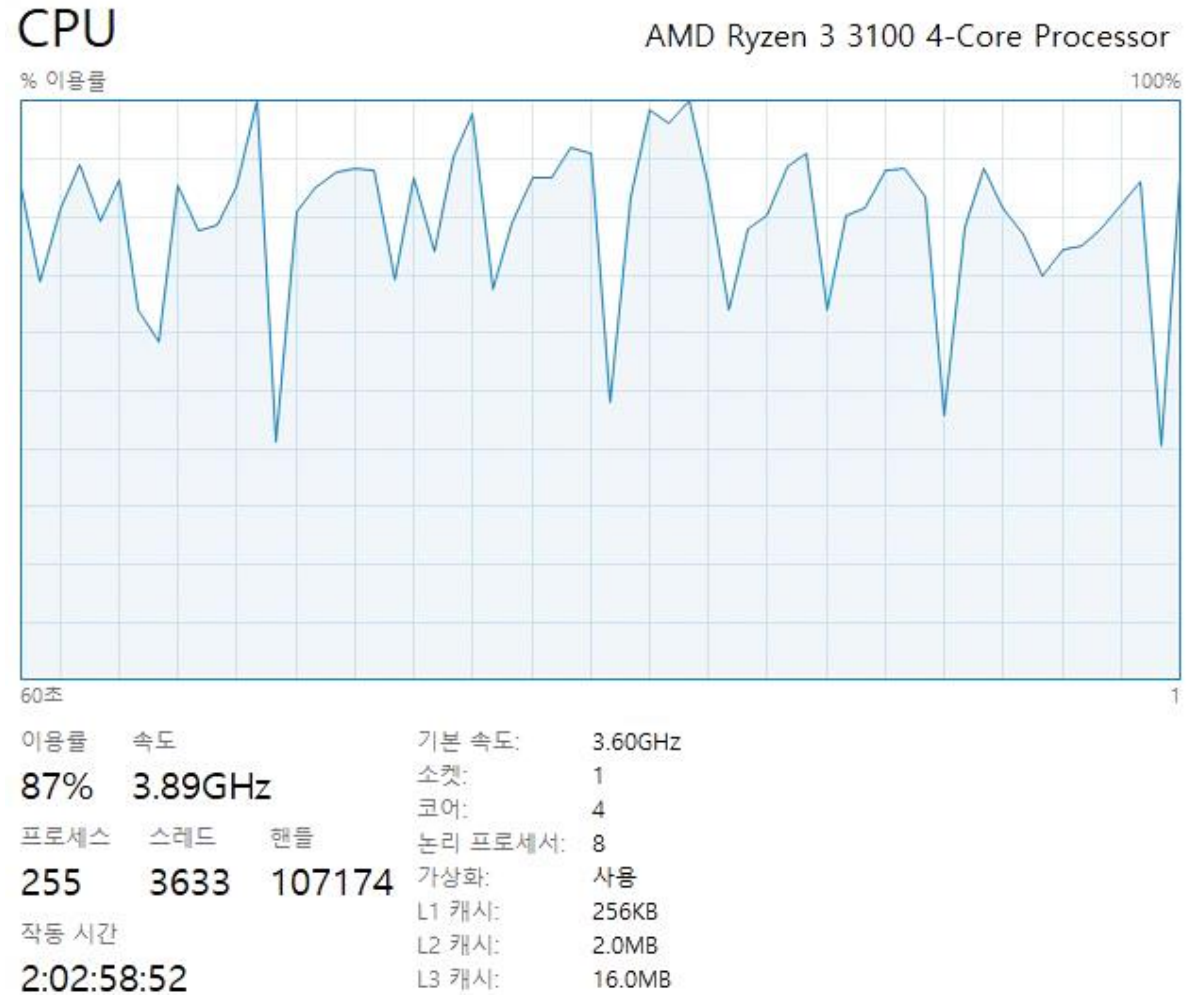
## CUDA 설정 오류

```
File "C:\Users\0310k\Anaconda3\envs\tensorflow36\lib\imp.py", line 243, in load_module
    return load_dynamic(name, filename, file)
File "C:\Users\0310k\Anaconda3\envs\tensorflow36\lib\imp.py", line 343, in load_dynamic
    return _load(spec)
ImportError: DLL load failed: 지정된 모듈을 찾을 수 없습니다.

Failed to load the native TensorFlow runtime.
```

## 2. Model

### Trial and Error





## 2. Model

### Experiments

#### ImageNet

20만 장의 이미지 만으로 실험

224 x 224 Resize

EfficientNet-b0 / Unet



Results



## 2. Model

### Experiments

#### ActivityNet

5초당 1프레임씩 이미지 추출

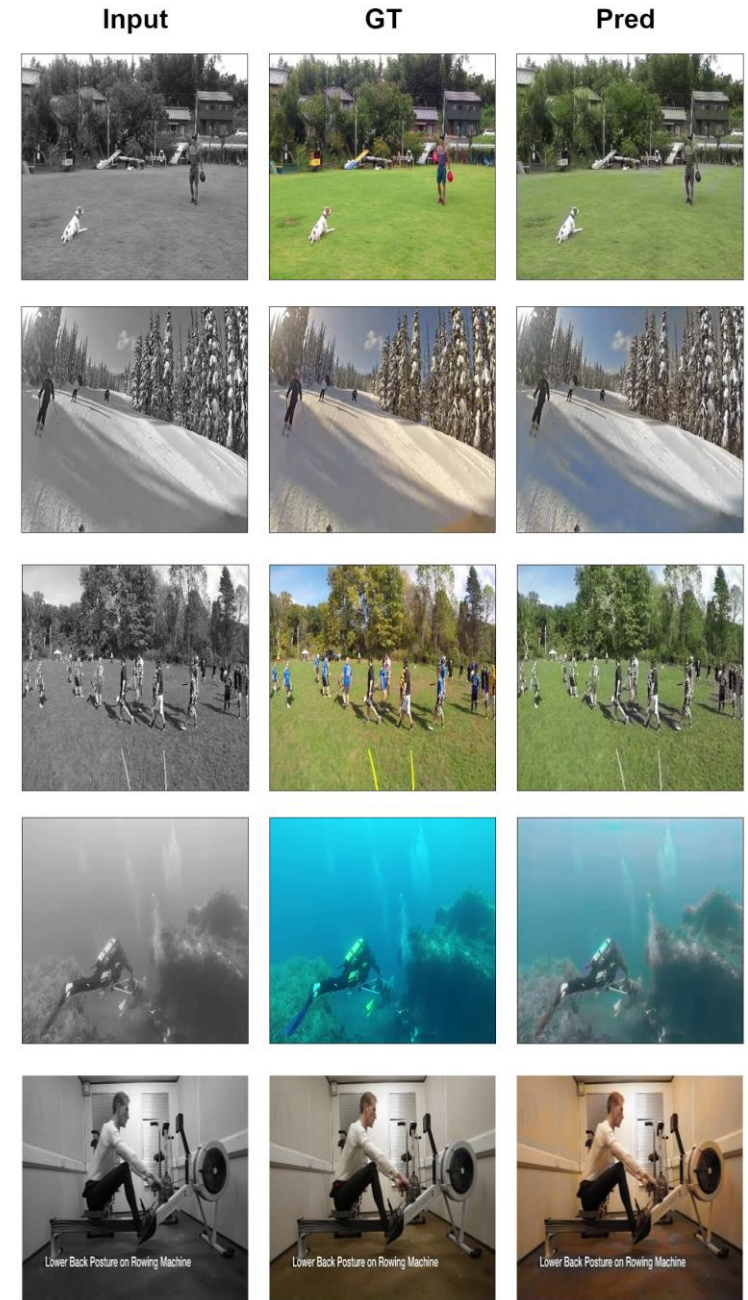
480px 이상의 해상도만 선택

Video\_id를 기준으로 GroupKFold

384 x 384 resize

17만장 이미지 데이터로 실험

EfficientNet-b0, b4 / Unet



Results

## 2. Model

---

### Trial and Error

384(input) -> 384(output)과 같이 비효율적으로 이미지 해상도를 사용하기 때문에  
학습 속도가 느리고, 비효율적임

현재는 patch 단위가 아닌 이미지 전체를 resize하여 학습하기에 작은 객체에 대한 성능이 낮음

## 2. Model

---

### Trial and Error

### Patch consistency

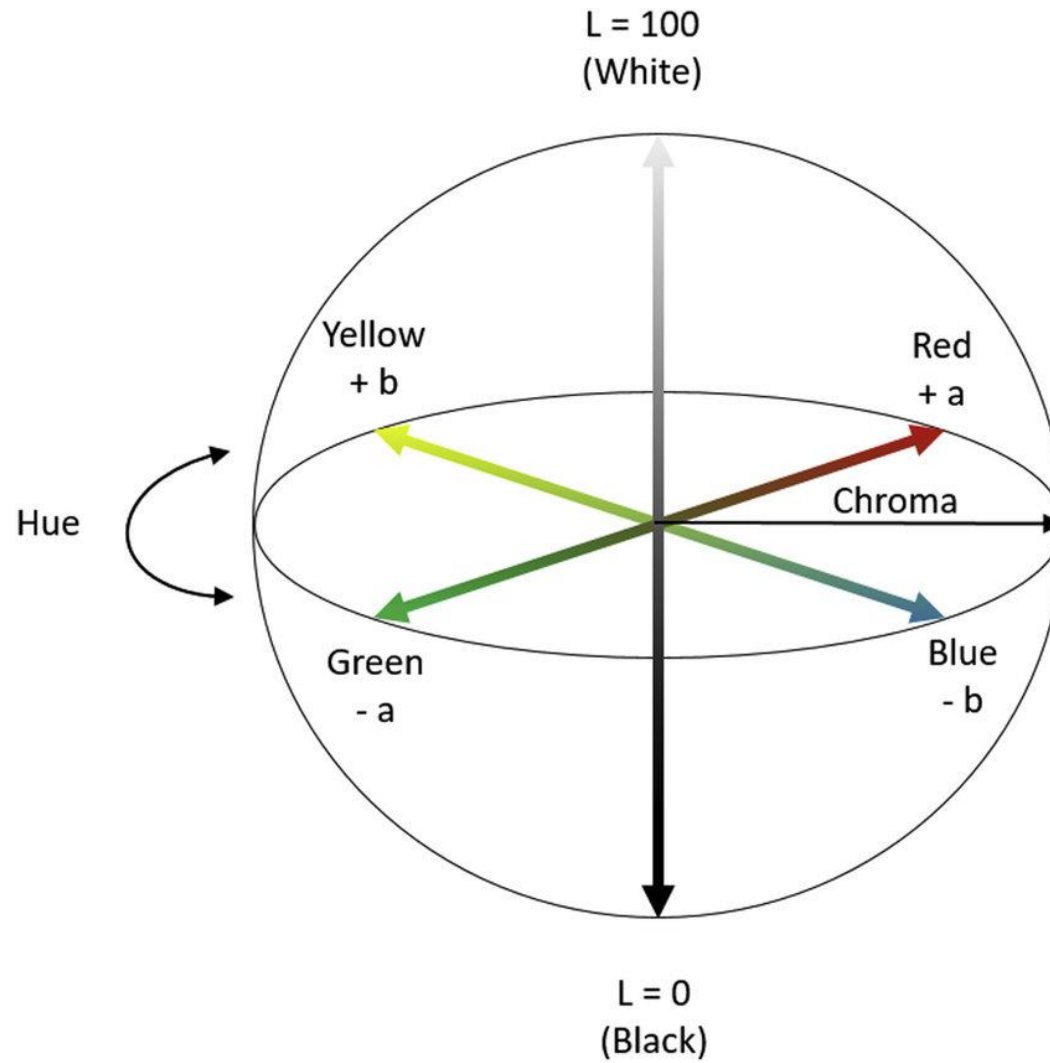
#### 학습 방법

- Regression : RGB, LAB / Classification : 3bits
- 기존 연구들에서 LAB로 transform 한 후에 학습을 진행하는 방법이 종종 보임
- 3 bits로 이미지를 quantization 해서 문제를 해결하는 경우도 있음 : ColTransformer

## 2. Model

### Trial and Error

LAB



## 2. Model

---

### Trial and Error

### 3-bit Quantization



감사합니다

---