

# Solution

## 프로젝트 수행 결과

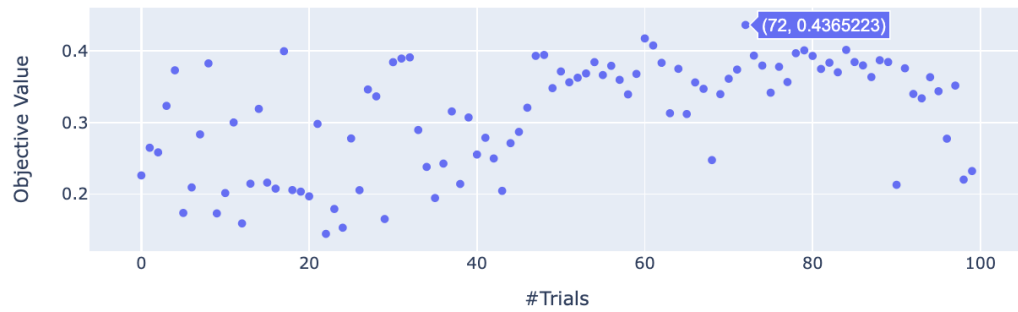
### ▼ Optuna tuning: (1차) 베이스라인

- 빠른 탐색을 위해 10 epoch, 전체 학습 셋의 40%만 사용
- Search Space
  - hyperparameter
    - optimizer = ['Adam', 'SGD']
  - data\_config
    - EPOCH = 10
    - SUBSET\_SAMPLING\_RATIO(train set 사용 비율) : 0.4
    - AUG\_TRAIN: randaugment\_train
      - n\_select : [0, 2, 4, 6]
    - BATCH\_SIZE = [16, 32]
    - IMG\_SIZE = [96, 112, 168, 224]
    - INIT\_LR = [1e-1, 1e-4], log uniform dist
  - model\_config
    - depth\_multiple = [0.25, 0.5, 0.75, 1.0]
    - width\_multiple = [0.25, 0.5, 0.75, 1.0]
- tune.py 로 탐색 종료 후 각 Objective Value plot을 살펴보고 적절한 탐색 모델을 선택하여 순차적으로 실험함
  - `study.best_trials` 에는 10개의 best trial이 저장되어 있는데, 그 순서대로 하지 않고 직접 값을 확인하여 선택
  - 후보 모델
    - `trial_id` : 72, 60, 4
    - `value_#` : f1, num\_params, inference time

	f1	n_params	inference_time(ms)	params_n_select	params_img_size	params_batch_size	params_depth_multiple	params_img_size
72	0.436522	389868.0	1.322323	0	168	16	0.25	168
60	0.417659	347852.0	1.276173	0	168	16	0.25	168
4	0.373051	326012.0	5.532058	0	168	32	1.00	168

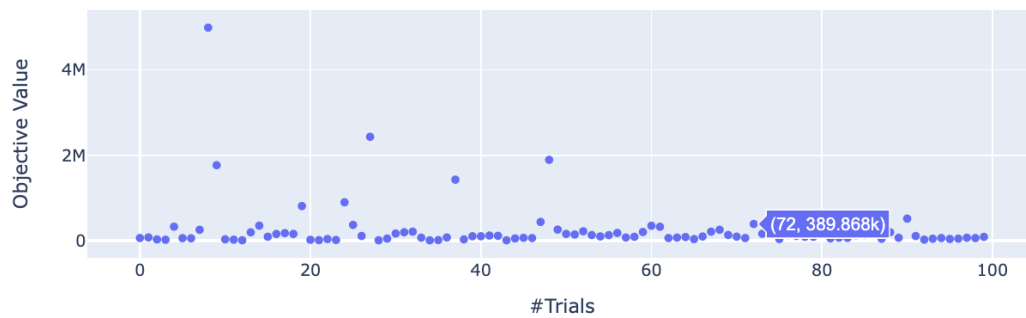
- F1 Score

Optimization History Plot



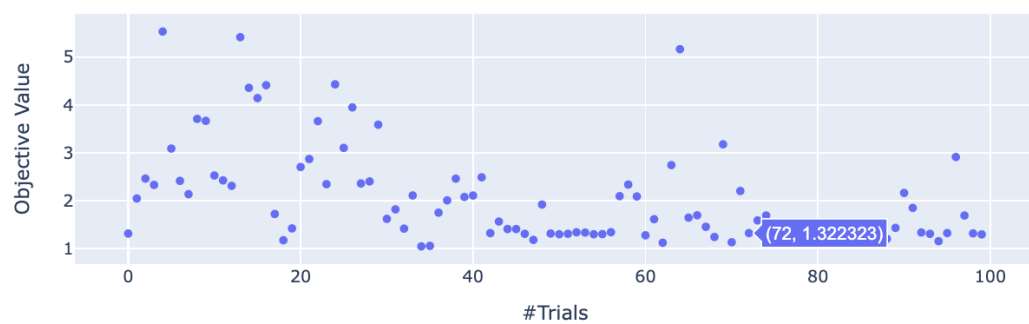
- Number of parameters

Optimization History Plot

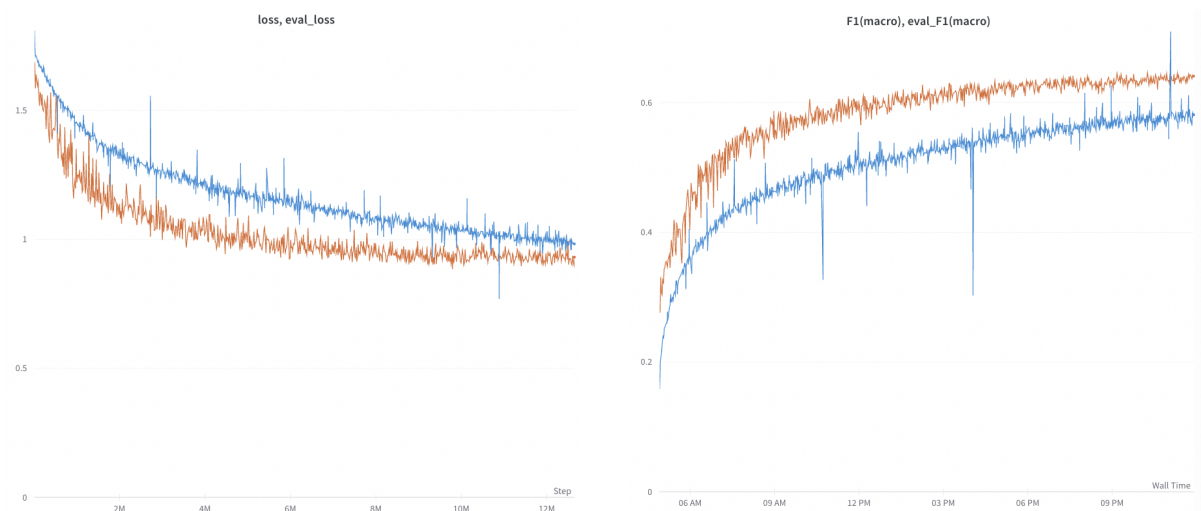


- Inference time

Optimization History Plot



- trial-id 72을 처음부터 다시 class weight를 적용하여 1000 epoch 동안 학습



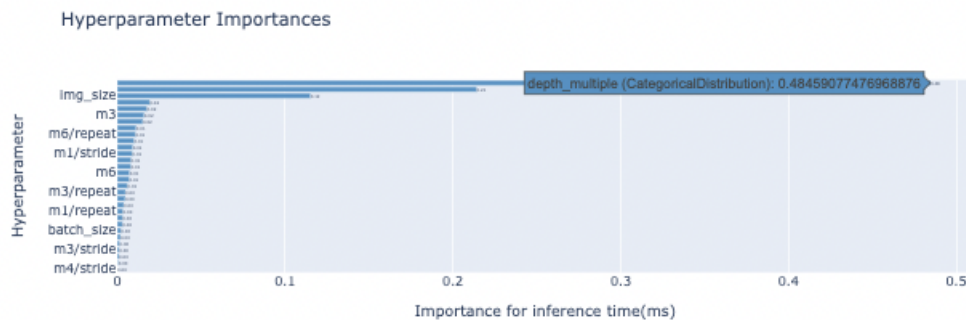
#### • LB 제출 결과

- local에서 측정된 시간이 약 1.4ms 보다 낮아야 제출시 inference time 60s보다 낮아지는 것을 확인
- f1 score는 validation set과 거의 유사
- n\_params = 389,868
- inference\_time(ms) = 1.3223

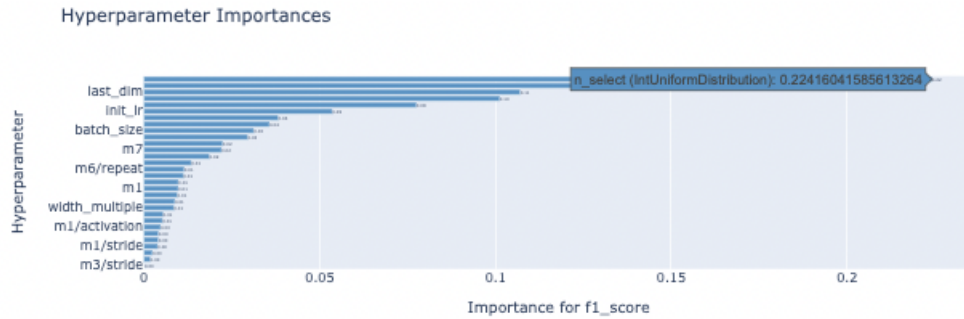
멤버	score (Rank)	f1	time	개요	생성 시간
허진규_T2242	1.4730	0.6170	54.5650	상세 보기	2021-11-29 23:28

#### ▼ Optuna tuning: (2차) 탐색 범위 재설정

- inference time에 가장 큰 영향을 주는 요소: depth\_multiple, m5 모듈의 종류, img\_size



- 성능에 가장 큰 영향을 주는 요소: n\_select, m6/stride, last\_dim, init\_lr

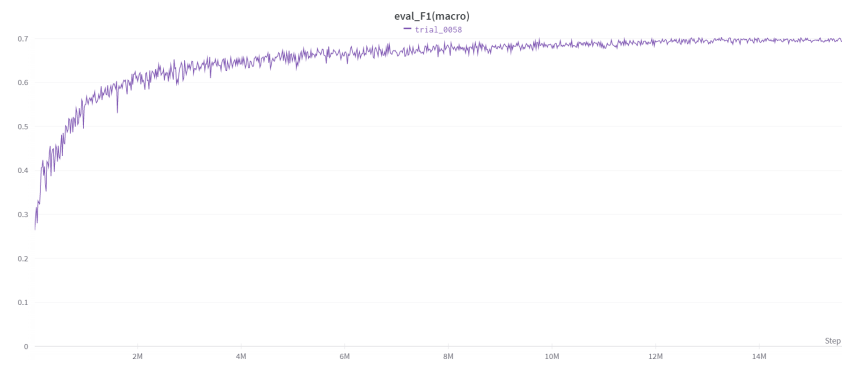


- 위 결과를 기반으로 탐색 범위를 아래와 같이 좁혀 재탐색
- Search Space
  - hyperparameter
    - optimizer = ['Adam']
  - data\_config
    - EPOCH = 10
    - SUBSET\_SAMPLING\_RATIO(train set 사용 비율) : 0.5
    - AUG\_TRAIN: randaugment\_train
      - n\_select : [0]
    - BATCH\_SIZE = [16, 32]
    - IMG\_SIZE = [168, 224]
    - INIT\_LR = [1e-4, 2e-3], log uniform dist
  - model\_config : 기본 제공 코드만 사용
    - depth\_multiple = [0.1, 0.3, 0.5]
    - width\_multiple = [0.6, 0.7, 0.8, 0.9, 1.0]
- 학습 모델 선택

number	values_0	values_1	values_2	datetime_start	datetime_complete	duration	params_batch_size	params_depth_multiple	params_img_size	..
16	16	0.405237	1586740.0	1.363699	2021-11-30 08:04:42.866444	2021-11-30 08:10:45.604608	00:06:02.738164	32	0.3	224 ..
17	17	0.416850	263260.0	1.455787	2021-11-30 08:10:45.625611	2021-11-30 08:16:32.481155	00:05:46.855544	32	0.1	168 ..
44	44	0.419979	310060.0	1.428139	2021-11-30 10:54:44.207745	2021-11-30 11:00:31.624101	00:05:47.416356	16	0.1	168 ..
57	57	0.409185	663436.0	1.353865	2021-11-30 12:17:29.866141	2021-11-30 12:23:20.494217	00:05:50.628076	16	0.1	168 ..
66	66	0.410768	58988.0	1.456196	2021-11-30 13:10:36.583716	2021-11-30 13:16:44.925041	00:06:08.341325	16	0.1	224 ..
67	67	0.428476	71700.0	1.433395	2021-11-30 13:16:44.943990	2021-11-30 13:22:53.815901	00:06:08.871911	16	0.1	224 ..

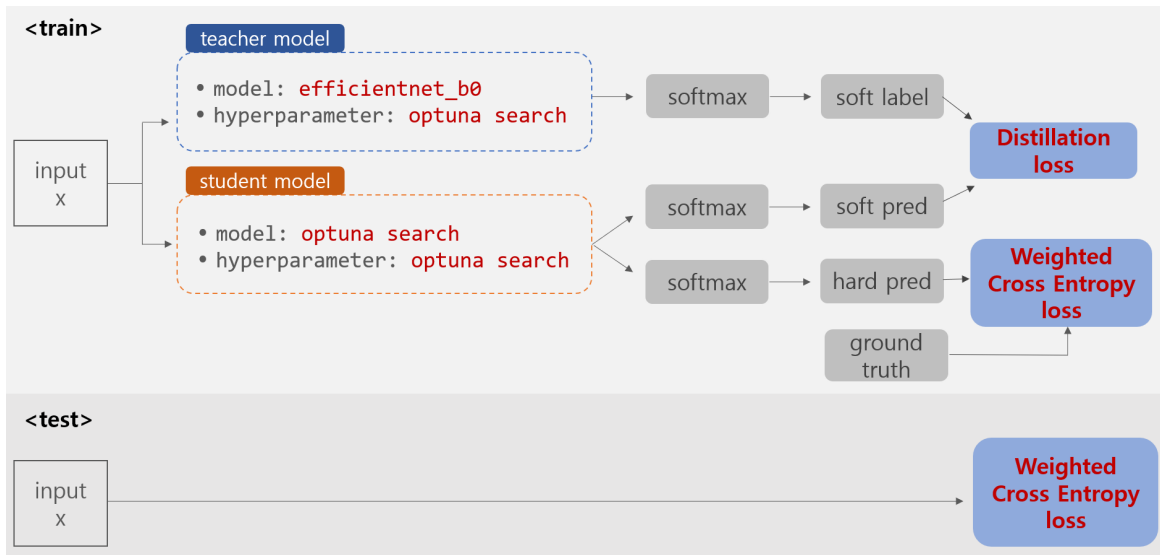
- 후보 모델

- `trial_id` : 44, 57
- `value_#` : f1, num\_params, inference time
- `study.best_trials` 에는 10개의 best trial이 저장되어 있는데, 그 순서대로 하지 않고 직접 값을 확인하여 선택
  - mean\_time (위 사진에서 values\_2) 이 1.4 이하
  - f1 score (위 사진에서 values\_0) 이 0.4 이상
  - depth가 낮고 width가 높을 때 성능도 많이 떨어지지 않고 빠른 속도의 모델이 나오는 경향을 보임
  - inference time 이 비슷하다면, 파라미터 수가 많은 것이 학습을 큰 epoch 으로 돌려도 쉽게 오버피팅이 발생하지 않을 것 같아서 파라미터 수가 많은 모델을 선택
- 학습 그래프



## ▼ Knowledge Distillation

- `teacher model` : timm의 **efficientnet\_b0** 모델에 대해 optuna를 활용하여 **hyperparameter search**하여 모델 tuning
- `student model` : optuna를 활용하여 **모델 및 hyperparameter search**하여 모델 tuning (4.2에서 tuning한 모델)
- Knowledge Distillation 전체 학습 과정
  - `train` : teacher model의 결과를 학습하기 위한 **Distillation loss**와 ground truth를 학습하기 위한 **Weighted Cross Entropy loss** 사용
  - `test` : **Weighted Cross Entropy loss** 사용하여 결과 도출

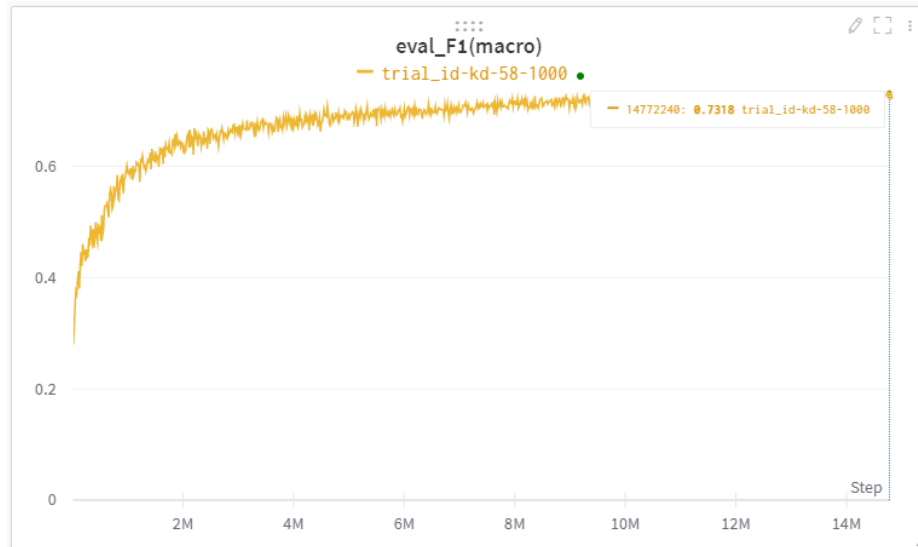


- efficientnet-b0 모델 학습

- 초록색 epoch=14로 설정하였을 때, overfitting 되었으며, 갈색 epoch=8로 설정하였을 때, 최고 성능 도달



- knowledge distillation 학습 결과



#### • 결과 분석 및 후기

- 학습 그래프에서 f1 score 0.73 도달
- wandb f1 score와 리더보드 score가 거의 유사하므로, 본 모델 제출에 성공하였다면 더 높은 score 획득 가능했을 것
  - 서버 제출 시, 서버 용량 및 dependency 문제로 failed가 떠서 최종 제출에 실패함. 시간 여유가 더 있거나, 미리 컴퓨터 환경을 체크했으면 좋은 성과로 이어졌을 것

#### ▼ 최종 모델 제출

- 최종 모델 구조

```
input_channel: 3
INPUT_SIZE: [168, 168]
depth_multiple: 0.1
width_multiple: 0.9
backbone:
- - 3
- Conv
- [48, 3, 1, null, 1, Hardswish]
- 2
- Conv
- [80, 3, 2, null, 1, ReLU]
- 4
- Conv
- [112, 3, 1, null, 1, Hardswish]
- 1
- DWConv
- [208, 3, 2, null, ReLU]
- 4
- Conv
- [176, 3, 2, null, 1, Hardswish]
- 1
- InvertedResidualv2
- [80, 6, 2]
- 1
- Conv
- [640, 1, 1]
- 1
- GlobalAvgPool
- []
- 1
- FixedConv
- [6, 1, 1, null, 1, null]
```

